BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi

Issue 80 | April 2019 | magpi.cc

The official Raspberry Pi magazine

THE 50 BEST RASPBERRY PI TIPS

ESSENTIAL HACKS, HINTS & TRICKS!

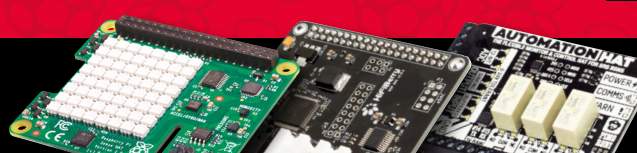
HACK LEGO BOOST

Take control of Lego with Raspberry Pi

PLUS!

52 PAGES OF PROJECTS & GUIDES

TOP 10 Raspberry Pi HATs

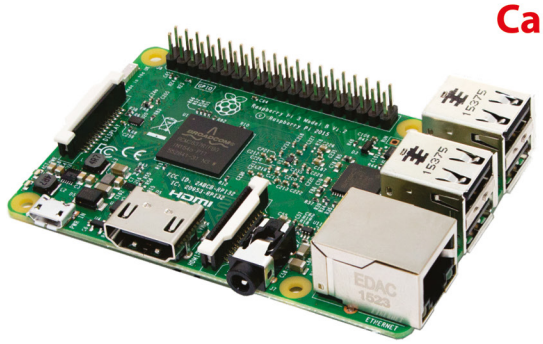


£5.99

Raspberry Pi PRESS

GLOBAL DELIVERY magpi.cc/store

CanaKit™



CanaKit Raspberry Pi 3 Ultimate Starter Kit

Model B | 1 GB RAM | 1.2 GHz | Quad-Core CPU

> Learn to Code
> Explore Computing
> Get started with Electronics

KIT INCLUDES RASPBERRY PI 3 AND ...

- **PREMIUM CASE & HEAT SINKS**

- **2.5A POWER ADAPTER**

- **32 GB CLASS 10 MICROSD CARD**

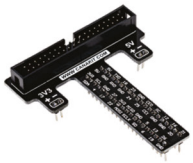
PRE-LOADED WITH OPERATING SYSTEM

- **USB MICROSD CARD READER**

- **PREMIUM HDMI CABLE**

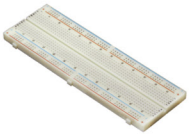
- **QUICK-START GUIDE**

- **GPIO TO BREADBOARD INTERFACE BOARD**



- **RIBBON CABLE**

- **FULL-SIZE BREADBOARD**



- **JUMPERS**

MALE TO MALE & MALE TO FEMALE

- **LEDS**

- **RESISTORS & PUSH-BUTTONS**

Available for worldwide shipping at:

WWW.CANAKIT.COM

Raspberry Pi Zero W

Now available at CanaKit!

Wi Fi™ Bluetooth®

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO

WELCOME
to *The MagPi* 80

This month we have the 50 Best Tips for Raspberry Pi. That's a bold claim, but we think our lead feature (page 26) delivers. *The MagPi* team worked together to bring you the best hacks, tricks, and tips. The result is a compelling list of tricks with something new for everybody. We all learnt something putting this feature together – and we're sure you will too.

This is the first in a series of articles designed to be useful for everybody, from beginners to seasoned experts. They're 'core features' that we're trying to develop to ensure *The MagPi* is accessible to all.

We're aware that some of our long-term readers want a little more to get their teeth into. That's why on page 68 you'll find our ham radio projects. We think ham radio and Raspberry Pi go hand-in-hand.

Meanwhile, Dave Honess – our favourite space buff – goes radio gaga with an amazing Pictures from Space tutorial (page 62). With a radio receiver, you can get pictures from audio files beamed directly from the International Space Station as it passes overhead. We love it!

That's just one of many cool tutorials this month. PJ has written about using a TV HAT to convert video files into comic books (page 54); KG has built a karaoke machine (page 58); and Mike Cook is busy hacking the new Lego Boost (page 38).

We're super-happy with the way this issue has come together. It's packed with great ideas from the whole team. We hope you love it as much as we do.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Editor of *The MagPi*. Lucy codes, crafts, and creates wonky robots. She speaks French (badly) and mangles the piano. One day she'll get that pet dog.

magpi.cc

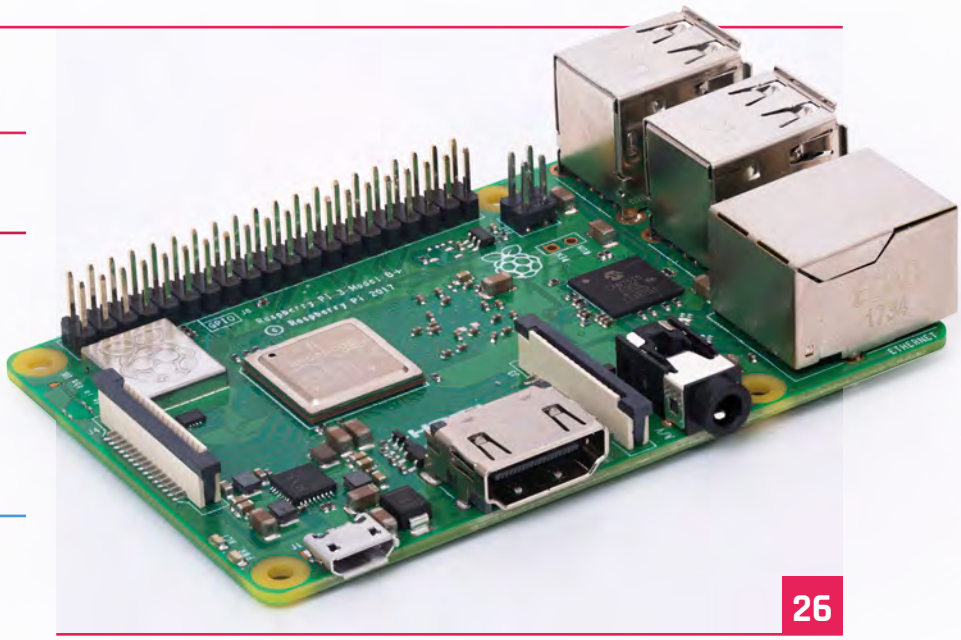
GET A RASPBERRY PI 3A+ WITH A SUBSCRIPTION! PAGE 24

Contents

Issue 80 > April 2019

Cover Feature

26 Tips & tricks



26

Regulars

06 The world of Pi
92 Your letters
97 Next month
98 Final word

Project Showcases

12 A Whisper of Moths
16 E-ink Calendar
18 Underwater Drone
20 YogAI
22 Clapper

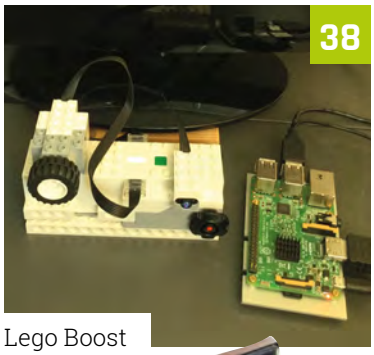
18

Underwater Drone

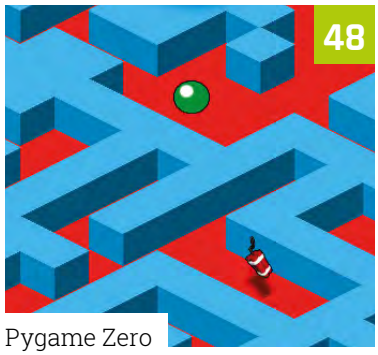
12

A Whisper of Moths

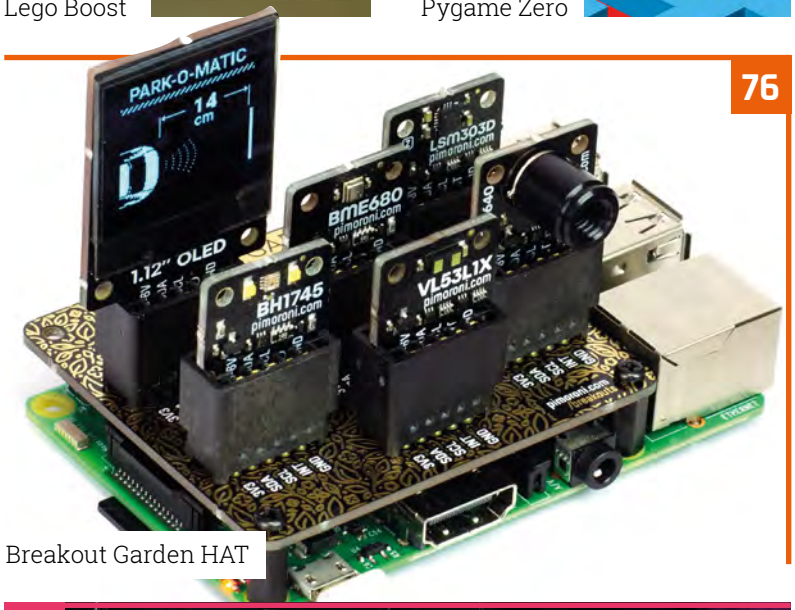
DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.



Lego Boost



Pygame Zero



Breakout Garden HAT

Claire Pollard interview

Tutorials

38 Lego Boost – Pi Bakery
44 Samba file server
48 Pygame Zero: AmazeBalls pt 3
54 Comic creator
58 Karaoke machine
62 Pictures from space

The Big Feature

Ham Radio Projects

Reviews

76 Breakout Garden HAT
78 Zero Dongle
80 Top 10 HATs
82 Learn the command line

Community

84 Claire Pollard interview
86 This month in Raspberry Pi
90 Events

WIN ONE OF **FIVE**

KEYBOW

Mechanical keyboards

GPIO Zero 1.5 launched

Be more in control of your Python projects with the new version of GPIO Zero. By **Rosie Hattersley**

GPIO Zero is a Python library with useful bits of code that make it easier to control Raspberry Pi-powered physical hardware: LEDs flash, buzzers buzz, and wheels turn. It's designed to make the Pi more accessible, especially for use in education.

Becoming a GPIO Zero hero has just become a whole heap easier thanks to the launch of an updated library of code, taking GPIO Zero to version 1.5.

This is great news if you want to start using your Pi for physical computing projects such as robots or home automation, since there's now more guidance on how to get going.

Lots of kits that previously took a fair amount of coding are now natively supported. Step forward the Halloween-themed Pi board, Xmas Tree with flashing LEDs, and the Jam HAT – ModMyPi's traffic light HAT with a piezo buzzer and buttons.

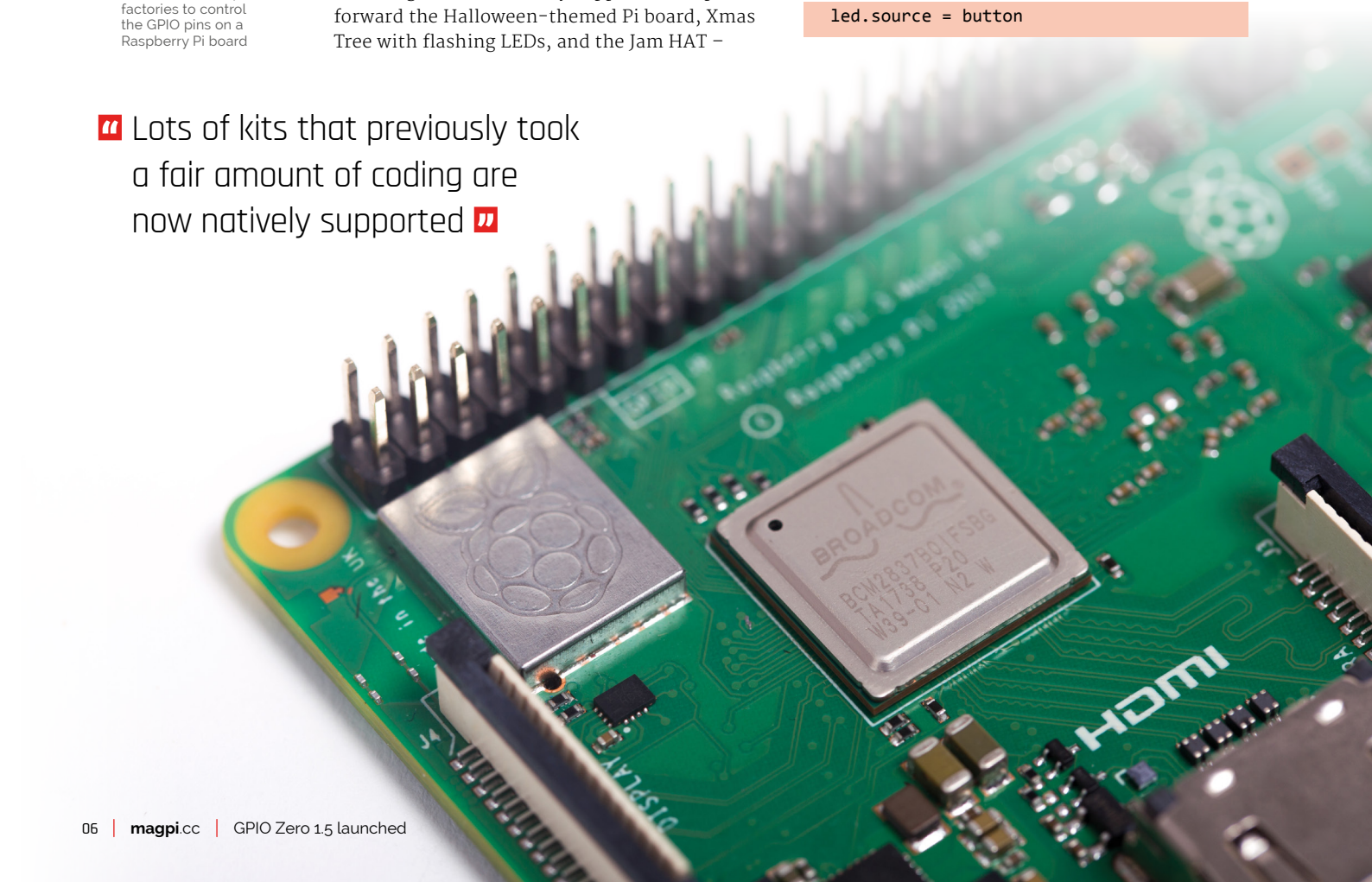
For audible amusement, try the tweaked version of Claire Pollard's tonal buzzer. GPIO Zero creator Ben Nuttall says the new software update includes the TonalBuzzer class, as well as a dictionary to help you choose suitable MIDI notes and frequencies.

Pairing devices

There's a neat way to connect devices together using callback functions. "In GPIO Zero v1.5, we've made connecting devices even easier," writes Ben. You can now use the following method to pair devices together:

```
led.source = button
```

> Lots of kits that previously took a fair amount of coding are now natively supported



▼ Advanced makers can create their own pin factories to control the GPIO pins on a Raspberry Pi board

As well as the standard pin factories (RPi.GPIO, pigpio, RPIO, and native), seasoned users will enjoy GPIO Zero's support for custom factories.

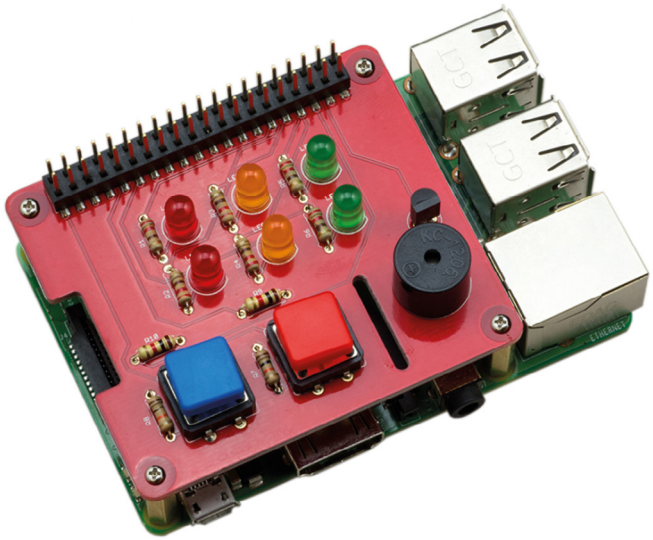
For instance, Google has written its own pin factory for the AIY Vision Kit (**magpi.cc/imfAQZ**).

This means you could use the pigpio library to remotely control individual pins and sensors. If you've struggled with the DistanceSensor class in the past, update GPIO Zero to v1.5 and use the pigpio pin factory for greater sensing accuracy.

To update GPIO Zero to the latest version, enter the following Terminal commands:

```
sudo apt update
sudo apt install python3-gpiozero python-gpiozero
```

For a complete list of what's new in GPIO Zero 1.5, see the changelog: **magpi.cc/vgTmJj**.



▲ GPIO Zero offers direct support for many popular hardware kits, such as ModMyPi's new Jam HAT

DAC+ ADC

Play back your music. And record it.

HiFiBerry

Up to 192kHz/24bit

Connect existing audio equipment and stream music over the network

Create a Karaoke box

Use the Raspberry Pi as an effects processor

Record audio

Create a web radio station

www.hifiberry.com

Smart calendar built with Raspberry Pi

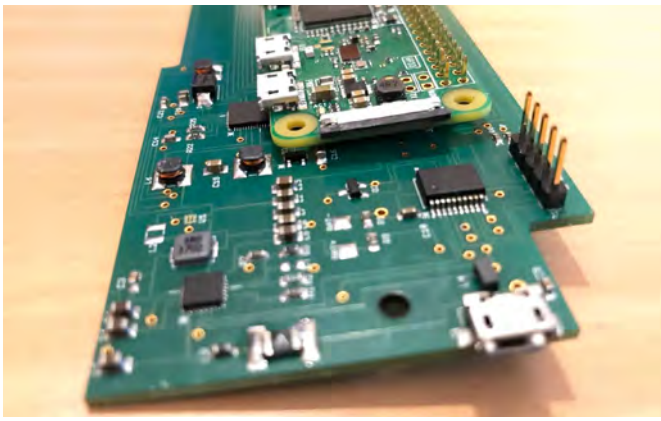
Vitec GmbH is hoping to sell a smart calendar with a Raspberry Pi at its heart. By **Lucy Hattersley**

Smart calendars are a hugely popular Raspberry Pi project, but not everybody wants to make their own. Some people just want to buy things.

One company hoping to bring a Raspberry Pi–based calendar device to the masses is Vitec GmbH, with its Smart Calendar Thync (**vitec-visual.com**).

"Managing the daily schedule, full of appointments and tasks, is a nightmare for most families," writes Marc Roset, CEO and Founder of Vitec GmbH. "Many businessmen and professionals find the compatibility of work and family calendars almost impossible."

The Smart Calendar Thync supports the synchronisation and connection of up to five different appointment calendars. Inside the case is a Raspberry Pi Zero W powered by a battery, running a 10.1-inch TFT display.



▶ Inside the Smart Calendar is a custom circuit board with a Pi Zero mounted on top of it

▼ Smart Calendar Thync uses a Raspberry Pi to bring a smart calendar device to the masses

Interestingly, the Smart Calendar Thync features a gesture sensor interface. "This allows you to quickly and easily switch back and forth by moving your hand. A short upward movement allows you to display further information about your current function," says Marc.

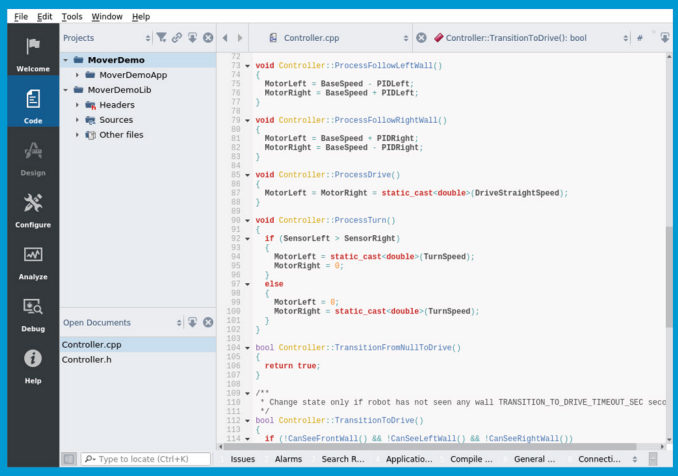
> "The Smart Calendar Thync features a gesture sensor interface"

"The simplest way is set up the product is via our app," he adds. The company has iOS and Android apps. "We don't use voice control for our product because a lot of people fear that other people can hear what they are saying."

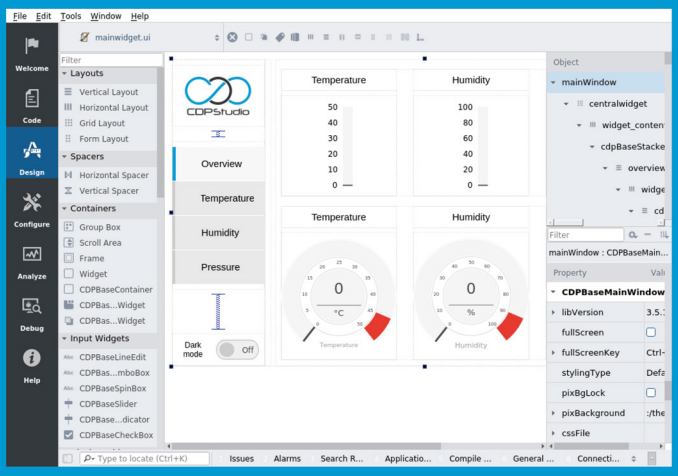
Light sensing

The Smart Calendar Thync also features an ambient light sensor, which measures the brightness of the surrounding area, regulates the brightness, and optimises energy usage.

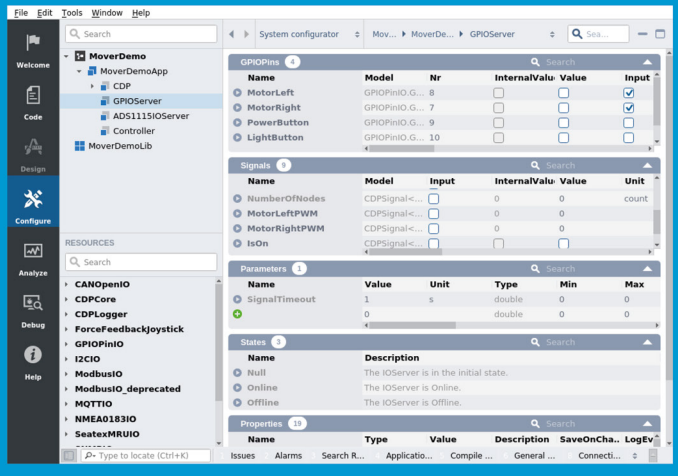
Users are able to mount the Smart Calendar Thync on a wall or use the stand to place it on a flat surface. A Kickstarter campaign for Smart Calendar Thync starts on 7 April 2019.



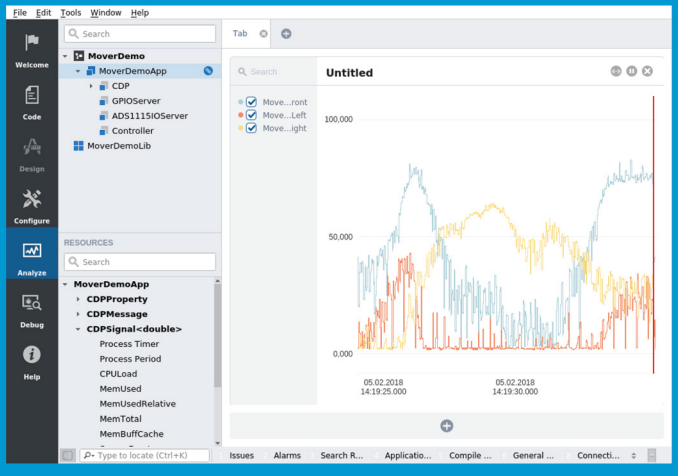
Code



Design



Configure



Analyze

Now free for home projects

A professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

Free download on www.cdpstudio.com

CDP Technologies AS
Nedre Strandgate 29
P.O. Box 144
NO-6001 Ålesund, Norway

Tel: +47 990 80 900
info@cdptech.com
www.cdpstudio.com

CDPStudio

3 ISSUES FROM £5

on a quarterly subscription

Use the code MP-SAVE at checkout

Visit: magpi.cc/345

Autism-friendly Code Clubs

Raspberry Pi has always been about making computing accessible; autism-friendly Code Clubs seem a natural step, believes **Rosie Hattersley**

When the original Raspberry Pi was born, back in the distant days of 2012, it was envisaged as a way of embracing physical coding and computing in a world that had become very app-based. Its central premise was to allow anyone access to a computer and computing, with cost a minimal barrier.

The lauded aims of Raspberry Pi and its focus on giving school-age children, in particular, coding and programming experience were enthusiastically taken up.

One major programme run by the Raspberry Pi Foundation is Code Club (**codeclub.org**), which works with a global community of volunteers, educators, and partners to run free coding clubs where 9 to 13-year-olds build and share their ideas.

Code Club says, "We think all children should have the opportunity to learn to code, no matter who they are or where they come from." There are 13,000 Code Clubs worldwide. Many of those in the UK are school-based, while others take place in libraries or youth clubs.

▲ The club is small but growing, with six children aged between 9 and 14 attending the sessions

◀ Volunteers engage the children in a variety of activities, including Scratch projects and physical digital making opportunities

However, as an after-school activity, Code Clubs are often lively and noisy, which isn't a comfortable environment for some children.

In the summer of 2018, Nicola Tubbs from Kent Libraries and Tim Cook from the National Autistic Society, Dartford and Gravesham branch, discussed the idea of a Code Club that would be accessible to children on the autistic spectrum. Together they planned and helped promote a bi-monthly, autism-friendly Code Club at Gravesend Library (**magpi.cc/YDreMb**) that now welcomes a regular group of 9- to 14-year-olds (**magpi.cc/GRioNB**).

Parents are welcome too, if the group members wish. Club members enjoy a quiet space, away from the main part of the library, where they can learn Scratch and work on physical computing projects. The club is hosted by Nicola, Tim, and volunteers from Digital Dens.

A Whisper of Moths

An art installation filled with hand-crafted moths that move. **Rob Zwetsloot** looks around it

MAKER

Macclesfield Community ArtSpace

A charity based in Macclesfield and run entirely by volunteers, it's a free workshop full of multitalented members able to create A Whisper of Moths.

magpi.cc/WNeWNR

We always love to see art installations that make use of the Raspberry Pi, and A Whisper of Moths is a perfect example. This particular project involved several people from the Print Mill, which is part of Macclesfield Community ArtSpace, along with technical and labour support from IDST! (If Destroyed Still True!), another ArtSpace group.

"The moths were made by local schoolchildren using recycled A4 plastic wallets that were drawn on and/or filled with glitter, ironed between baking parchment and cut to shape," Nick Young from the team tells us. "Some moths were also 3D-printed."

The movement of the moths was simulated by projecting randomly positioned and sized circles onto the moths, which were hanging by fishing line from fine garden netting suspended from the church balcony.

The moths were made using recycled, or by recycling, materials

An original Raspberry Pi was recycled to act as the audio player, sensing motion to trigger it

The installation was interactive, with the sound of whispers being triggered as people walked through, along with the projected circles of light to simulate movement to some degree. How did such an idea come about, though?

Lunar new year

"The Town Council approached Macclesfield Community ArtSpace wishing to celebrate in some form, Chinese New Year," Nick explains. "Macclesfield's link with China is through silk and therefore the suggestion was made to create an installation that incorporated silk moths and engaged the local schoolchildren in the making process. Macclesfield is also striving to draw attention to the problems of single-use plastic and so we chose to use plastic and recycle it."

The main attraction is the display of moths created by school students

The Pi Zero helped control the random shapes on the projector

▲ Even without the nice stained-glass light, the butterflies themselves are very colourful

▶ The project setup was quite involved – a lot of the church was used!

Quick **FACTS**

- Using an original Raspberry Pi Model B was part of the recycling efforts
- A lot of the project was programmed in Pygame
- The sounds have been reused for Halloween and Christmas
- We're currently in the Year of the Pig, hence the papier mâché pig
- Macclesfield was once the world's largest producers of finished silk

Whispering to moths

01 As visitors walk into the church, the PIR sensor picks them up and begins the A Whisper of Moths sequence.

02 The original Pi Model B begins to play a random sample of whispering sound effects, which is how the project gets its name.

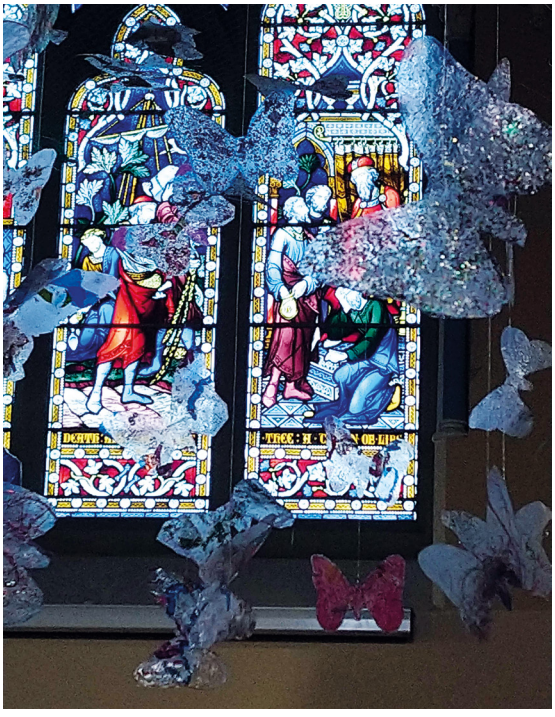
03 As the audio plays, random circles of light are projected onto the moths hanging above the visitors, which – along with the whispering – gives an illusion of movement.

▲ A dragon dance led the visitors to the church, where they would experience A Whisper of Moths

After following a Chinese dragon puppet and papier mâché pig down the high street in a parade during the Chinese New Year festivities, people walking into the church would trigger the display. "We had the added and unforeseen bonus of coloured light filtering through the moths as the sun shone through the huge stained glass windows," Nick recalls.

Pi connection

"We discussed motorising some or all moths, even just with vibration motors, but discarded this as inappropriate for a sustainable art-piece," Nick says. "We decided to have a moving pattern of light [shone] around or onto the static exhibit to create the illusion of movement. We created a prototype using metres of LEDs, but these were not bright enough and on surveying the venue, an active local church, we realised that we could use a projector to beam the light onto the art.



▲ Each butterfly was painstakingly added to the net

◀ The stained glass windows added some unexpected colour to the moths

> We realised that we could use a projector to beam the light onto the art

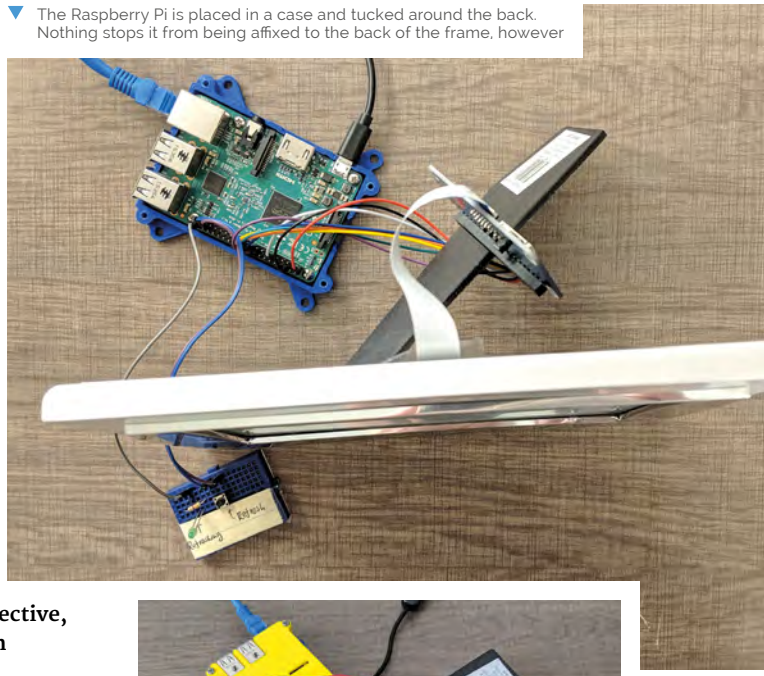
"We used the first Raspberry Pi for the sound because the simple movement sensor could trigger and play many sound files, reusing an existing setup, and a second Raspberry Pi Zero also programmed with Pygame to display a series of random white circles against a black background."

The final setup was pretty simple, making use of an old Pi Model B, a Pi Zero W, and PIR motion sensor, plus the speakers and projection equipment.

Currently, the exhibition has been taken down. However, the team are looking to set it up in a local museum soon.

E-ink Calendar

Zonglin Li has proven e-ink displays are not just for reading by creating a smart calendar hub. **David Crookes** takes a look



The Raspberry Pi is placed in a case and tucked around the back. Nothing stops it from being affixed to the back of the frame, however

MAKER

Zonglin Li

Zonglin is a senior student at University of Illinois, majoring in Computer Science and Psychology. Trained to be a software engineer, he also loves tinkering with cool hardware.

magpi.cc/WwAoZP

Some project ideas are simple yet effective, allowing makers to get creative with otherwise unused items to produce something practical or fun. In Zonglin Li's case, inspiration struck when he was attempting to find a use for an e-ink screen he'd bought a long time ago without a clear project in mind. "I came across the box containing the screen and it reminded me that I should do something with it," he says.

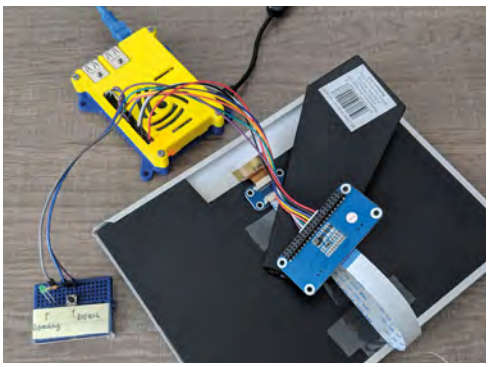
At first, he considered turning it into a straightforward digital clock. "But I soon figured that it was impractical due to the long refresh time – every minute would require a full update, which is visually annoying," he explains. He then began to think about the kinds of information which could tolerate a longer refresh period.

"The project was not really ambitious at the beginning and all I wanted initially was something that could show me the date and perhaps the weather," says Zonglin. From there, however, he had the idea of incorporating his daily activities, seeing the existence of the Google Calendar API as an ideal opportunity.

> "The e-ink display simplified the design, especially if you wanted to put the device in the bedroom"

Paper Pi

To start, he opted to use a Raspberry Pi because of its ability to run a full OS and its abundance of GPIOs: "Having a full OS meant that I could use high-level programming languages which saved a lot of time, especially when it came to Google OAuth and API integrations. The screen manufacturer also had nice documentation for Raspberry Pi, so it was easier to get started."



Some e-ink screens support rapid partial update with multiple grey scales, so Zonglin is considering upgrading to add real-time notification mirroring using Pushbullet

He also quickly realised that the e-ink screen would be a perfect fit. "I believed it would give a feeling of a real paper calendar that happened to display up-to-date information," he says. "The e-ink display also simplified the design, especially if you wanted to put the device in the bedroom, because it doesn't need to be backlit to operate. Unlike using LCD, you don't have to design the logic for adjusting the brightness either."

Widget designs

Even so, much of the effort went into coding, with Zonglin coming up with ideas for widgets before figuring how best to structure them. He composed a date widget from a matrix of text widgets while the weather widget used Unicode text with a nice font for its icon and integrated with OpenWeatherMap API.

He also designed a widget for events using Google OAuth and Calendar API, adding an

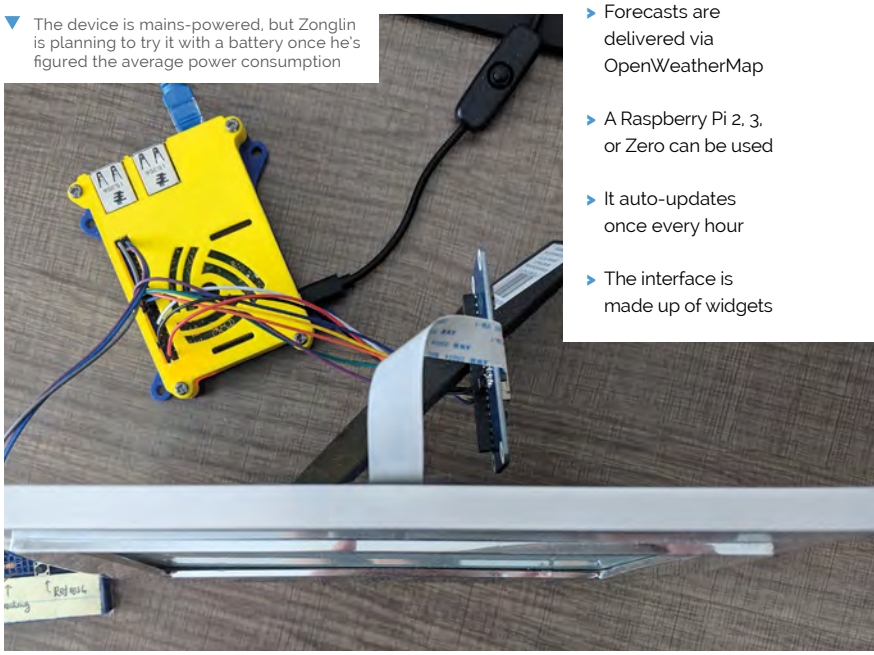
As well as forecasting the weather, the Google Calendar API displays the temperature and chance of rain
The widgets are rendered to a buffer (a PIL image) first before being copied to the screen
An LED lights up when the screen is being refreshed, and a manual button has been added too
Events
Mar 10, Sun Daylight Saving Time starts
Mar 17, Sun St. Patrick's Day
Apr 15, Mon Tax Day
Apr 21, Sun Easter Sunday
Apr 22, Mon Easter Monday
May 05, Sun Cinco de Mayo
May 12, Sun Mother's Day
May 27, Mon Memorial Day
February

interactive guide to create a config file and parser. Finally, he sketched out each widget on paper, specifying the relative location of each component, and worked out the best-looking sizes before laying out their locations on the screen.

"I tried my best to avoid hard-coding the offset of each widget," he reveals. "In this way, if I set the widgets to have a different size, all the components in the widget will be resized as well."

Once in place, Zonglin considered the look of the overall project, using the GPIOs for a manual refresh button while also allowing for automatic hourly updates. This ensured the information does not become not too stale, especially for weather.

To add some extra polish, the screen was placed in a 7.5-inch photo frame and the electronics – the Raspberry Pi, 330 Ω resistor, and breadboard – were tucked around the back. The result is a good-looking smart calendar. "The code can also be ported with minimal effort should I get a screen of a different size in the future," he says.



The device is mains-powered, but Zonglin is planning to try it with a battery once he's figured the average power consumption

Quick FACTS

- It uses a 7.5-inch Waveshare e-ink display HAT
- Forecasts are delivered via OpenWeatherMap
- A Raspberry Pi 2, 3, or Zero can be used
- It auto-updates once every hour
- The interface is made up of widgets

Underwater Drone

Ievgenii Tkachenko not only took the plunge with a Raspberry Pi but with his first Pi project too, as **David Crookes** explains

MAKER

Ievgenii Tkachenko

Ievgenii is a senior Android developer from Kyiv, Ukraine. As well as writing code, he loves engineering challenges and creating something interesting and helpful.

magpi.cc/GRSQHi

Never let it be said that some makers won't jump in at the deep end for their ambitious experiments with the Raspberry Pi. When Ievgenii Tkachenko fancied a challenge, he sought to go where few had gone before by creating an underwater drone, successfully producing a working prototype that he's now hard at work refining.

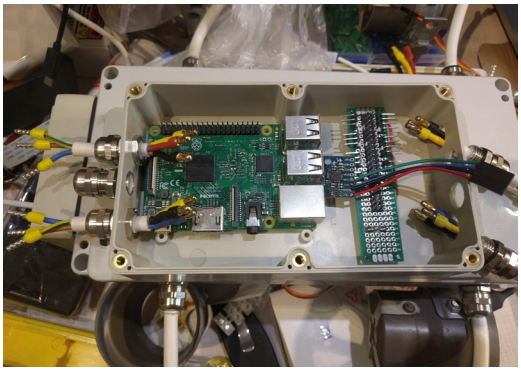
Inspired by watching inventors on the Discovery Channel, Ievgenii has learned much from his endeavour. "For me it was a significant engineering challenge," he says, and while he has ended up submerging himself within a process of trial-and-error, the results so far have been impressive.

Pi dive

The project began with a loose plan in Ievgenii's head. "I knew what I should have in the project as a minimum: motions, lights, camera, and a gyroscope inside the device and smartphone control outside," he explains. "Pretty simple, but I didn't have a clue what equipment I would be able to use for the drone and I was limited by finances."

Bearing that in mind, one of his first moves was to choose a Raspberry Pi 3B, which he says was perfect for controlling the motors, diodes, and gyroscope while sending video streams from a camera and receiving commands from a control device.

The LEDs are attached to radiators to prevent overheating, and a pulse driver is used for flashlight control



The Raspberry Pi 3 sits in the housing and connects to a LiPo battery that also powers the LEDs and motors

"I was really surprised that this small board has a fully functional UNIX-based OS and that software like the Node.js server can be easily installed," he tells us. "It has control input and output pins and there are a lot of libraries. With an Ethernet port and WiFi and a camera, it just felt plug-and-play. I couldn't find a better solution."

Working with a friend, Ievgenii sought to create suitable housing for the components, which included a twin twisted-pair wire suitable for transferring data underwater, an electric motor, an electronic speed control, an LED together with a pulse driver, and a battery. Four motors were attached to the outside of the housing and efforts were made to ensure it was waterproof. Tests in a bath and out on a lake were conducted.

Streaming video

With a WiFi router on the shore connected to the Raspberry Pi via RJ45 connectors and an Ethernet cable, Ievgenii developed an Android application to connect to the Raspberry Pi by address and port ("as an Android developer, I'm used to working with the platform"). This also allowed movement to be controlled via the touchscreen, although he says a gamepad for Android can also be used. When it's up and running, the Pi streams a video

Wires leave the housing through cable glands and sit in the water, but they have been sealed

The camera was placed in this transparent waterproof case attached to the front of the waterproof housing

The drone uses four brushless motors – two positioned horizontally and two vertically. Ievgenii would like to add two more verticals for better control

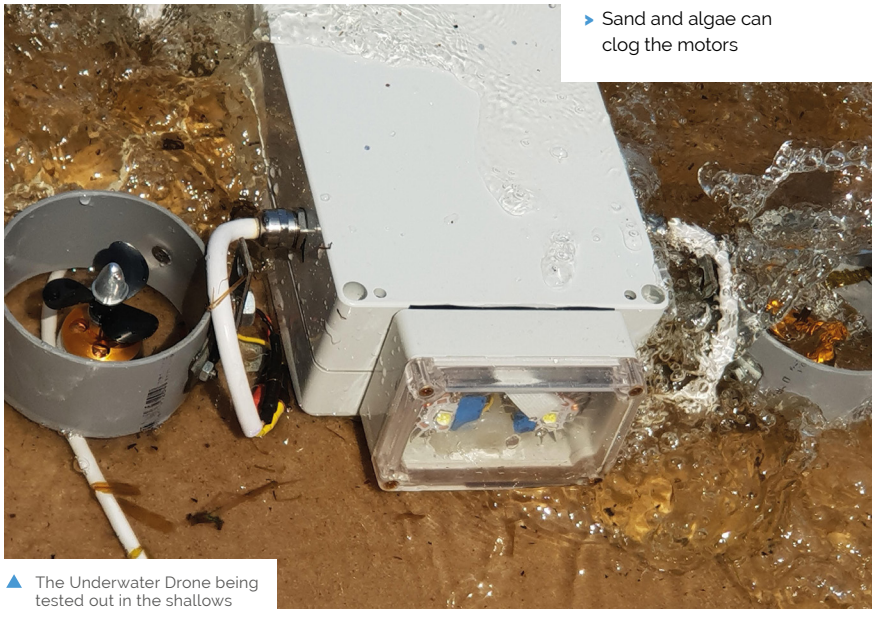
Quick FACTS

- The prototype worked well at a depth of two metres
- It can be controlled via an Android app
- A lot of testing was required
- It operated best in a calm lake
- Sand and algae can clog the motors

from the camera to the app – "live video streaming is not simple and I spent a lot of time on the solution" – but the wired connection means the drone can only currently travel as far as the cable length allows.

In that sense, it's not perfect. "It's also hard to handle the drone and it needs to be enhanced with an additional controls board and a few more electromotors for smooth movement," Ievgenii admits. But as well as wanting to base the project on fast and reliable C++ code and make use of a USB 4K camera, he can see the future potential and he feels it will swim rather than sink.

"Similar drones are used for boat inspections and they can also be used by rescue squads or for scientific purposes," he points out. "They can be used to discover a vast marine world without training and risks too. In fact, now that I understand the Raspberry Pi, I know I can create almost anything, from a radio electronic toy car to a smart home."



▲ The Underwater Drone being tested out in the shallows

YogAI

This marvellous magic mirror features an AI yoga instructor to guide you through your poses. **Phil King** limbers up

MAKER

Salma Mayorquin and Terry Rodriguez

A team of machine-learning practitioners, software developers, hardware tinkerers, and security enthusiasts interested in exploring real-world applications of AI.

smellslikeml.com

We've seen many 'magic mirror' projects over the past few years, featuring a TV screen behind the glass to show useful information, but YogAI (magpi.cc/nQTDgX) takes the concept to a whole new level by providing an AI personal trainer to guide and correct your yoga positions.

Self-confessed fitness-nuts Salma Mayorquin and Terry Rodriguez thought that having a personal trainer could be a way to keep track of their fitness progress, so why not try to make a virtual one? "With [deep learning] models like pose estimation, we figured there was a way we could make a program that could track how we were exercising and started experimenting from there," says Terry.

"YogAI guides users through a flow of yoga poses, offering generally helpful advice when the camera senses a user not in the correct pose," explains Salma. "At the heart, YogAI uses pose estimation to find reference key points on the body. This is used to understand and classify common yoga poses."

> A Raspberry Pi 3 interprets the camera images in real-time, detecting key body points to display the pose on the mirror and classify it

Users interact with YogAI through both visual feedback via the mirror display, and a voice interface – using the Snips AIR voice assistant (**snips.ai**) – which enables the user to give spoken commands to start, stop, pause, and restart a yoga session. YogAI also talks back through the Flite voice synthesizer to guide the yogi to achieve the correct poses.

▲ The AI has been trained to detect 14 poses, including three versions of the Warrior

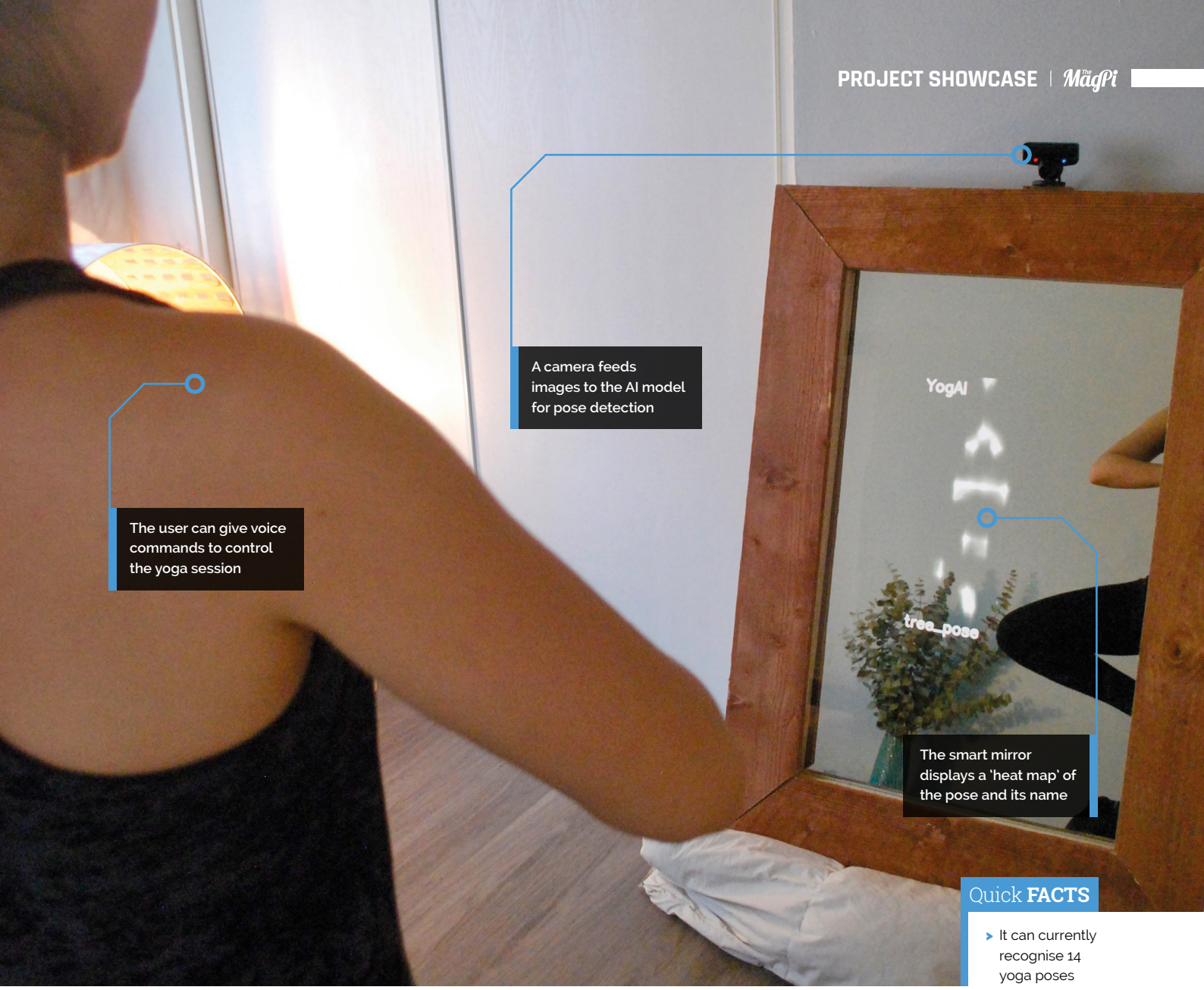
While a prototype magic mirror only took the experienced makers a week to build, training the AI to recognise yoga poses in real-time was a trickier task. "We need our computer vision models to run quickly so that we have enough resolution in time to identify the move," reveals Terry.

Strike a pose

A Raspberry Pi 3 interprets the camera images in real-time, detecting key body points to display the pose on the mirror and classify it using a deep-learning model trained with a dataset of around 35,000 samples.

However, the pair found that the Pi could only run image inference at one frame every 4–5 seconds, resulting in lag. A workaround was soon found: "Shrinking our pose estimation models down using TensorFlow Lite, we were able to bring our frame rate from 0.2 fps to 2.5 fps," says Salma. "For faster inference, we will look for ways to reduce the model further. We also believe upgrading to the Raspberry Pi Compute Module 3 will increase the performance significantly."

"Overall, the accuracy across a dozen common poses is roughly 80%," divulges Terry. "Not


A camera feeds images to the AI model for pose detection
The user can give voice commands to control the yoga session
YogAI
tree_pose
The smart mirror displays a 'heat map' of the pose and its name

surprisingly, we find similar pose variants, e.g. warrior poses, can be a source of confusion. When the head/face is blocked, the pose estimates degrade, which impacts our classification of poses like downward dog."

More intense exercise

As well as using the system for yoga, Salma and Terry are planning to adapt YogAI to monitor more energetic workouts. "We're interested in strength training, and others have suggested dance and karate katas," says Terry. "We think YogAI is well-positioned to perform more general health and personal wellness tasks."

"We want to integrate with popular health wearables," adds Salma. "A smart watch with an accelerometer and heart rate monitor can introduce a lot of important context to bring YogAI closer to our vision for a smart mirror yoga instructor and toward a personal wellness platform."

They used a 2×3 ft one-way mirror and built a wooden frame around it

Quick FACTS

- It can currently recognise 14 yoga poses
- TensorFlow Lite is used for the AI pose training
- The software is available on GitHub: **magpi.cc/emKnDd**
- Terry and Salma are experimenting with adding a life-like avatar
- Other Pi projects include a seizure prediction wearable (**magpi.cc/OsMEve**)

The Raspberry Pi Clapper

Your wishes fulfilled at the clap of your hands? **Nicola King** applauds a very clever creator

MAKER

Ash Puckett

Ash is an independent tech writer from Appalachia. When she isn't busy illustrating, she's knuckles-deep in a Raspberry Pi, creating awesome new projects to share with the world.

ashpuckett.com

Ash Puckett thought her life could do with a little simplifying, so she came up with a thunderclap of an idea, as she recalls: "I rolled out of bed one morning, clapped my hands, and remembered I don't have clap-powered lighting. What better way to remedy the situation than to create my own Raspberry Pi Clapper?"

What better way indeed? Ash then took a Raspberry Pi, along with a USB microphone, and created something elegantly simple, but useful at the same time. Amazingly, the entire project was conceptualised and completed in one day, then made into an online tutorial (**magpi.cc/CcMzdP**).

"The challenge is in creating a guide with flexibility," she reveals. "I wanted to create a starting point for anyone with a similar project in mind. Whether you want clap-powered music or a clap-powered garage door, there's a common foundation they all share, and I wanted to provide that."

Your clap is my command

So, how does it work exactly? "The Raspberry Pi is programmed to listen for claps using a USB microphone," explains Ash. "You can use this to trigger custom events. Pretty much anything you can control with Python can be manipulated using the power of clapping."

> I knew from experience the Pi was more than capable. You can really take advantage of the GPIO pins to create some awesome projects

The possibilities, in terms of what you can control, are vast. "I've been able to control an LED, initiate some funky tunes, and run a few custom scripts. The proof of concept has been demonstrated – from here it's a matter of imagination," says Ash. "If you can control it with a Pi, you can control it with claps. This includes home automation using relays, controlling smart kitchen appliances, and lighting."

Ash used the Raspberry Pi for her make because, "I knew from experience the Pi was more than capable. You can really take advantage of the GPIO pins to create some awesome projects. I wanted to use something that could handle a complex project, if desired. [...] The Raspberry Pi is perfect for this kind of creative endeavour."

Inspired yet? Ash has received some very positive feedback from the Pi community: "I've also seen people come forward with their own Clapper project ideas – which I adore. Plenty of people have proposed their own modifications, recommended specific hardware, and provided new avenues for Clapper projects to take root."

She adds, "Many people have sound-activated technology on their minds. It's not surprising – almost everyone has a voice-activated assistant in their pocket. As people realise they can use audio-based control for their Pi projects, we'll see a surge in those ideas. The Clapper project is just a small component of that movement."

Handy...crafter

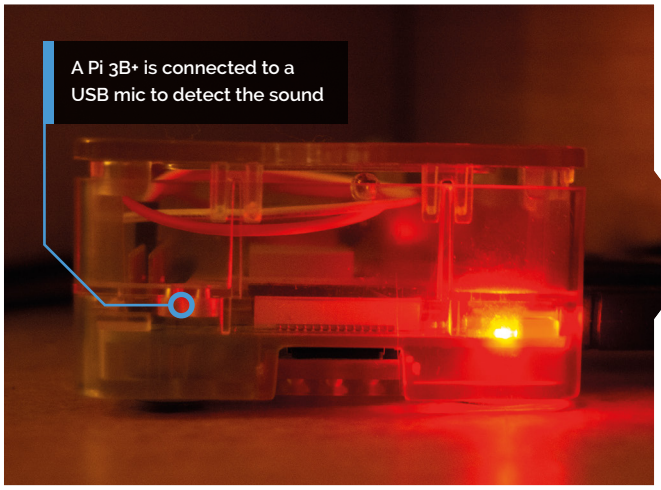
As to the future, Ash has plans aplenty in terms of what she hopes to develop. "I've got a whole list of Raspberry Pi projects ready to go – from home security systems to an automated greenhouse. These projects will use the Raspberry Pi as a centralized control system that can be remotely accessed and controlled. You can follow me on howchoo (**magpi.cc/Demoez**) to keep up with my latest Pi projects. And you better believe that if I can power them with claps, I will."

The simplicity of the Clapper arguably makes it a very appealing project. Give it a go yourself, or see what other clapping-connected ideas you can come up with! Send us your projects **@TheMagPi** on Twitter or email **magpi@raspberrypi.org**.

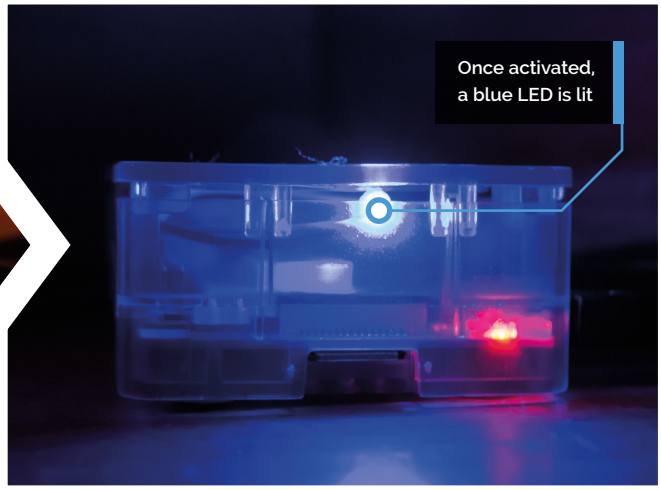
A simple clap of the hands is used to trigger the device

Quick FACTS

- Claps are detected using Nikz Jon's pi-clap script – **magpi.cc/GDpKhv**
- You can follow Ash's howchoo tutorial at **magpi.cc/CcMzdP**
- The PyAudio library is used to detect sound from a USB mic
- You could trigger many different items with a clap – use your imagination!
- Ash's other Pi projects include a smart alarm clock and a music-streaming device



A Pi 3B+ is connected to a USB mic to detect the sound



Once activated, a blue LED is lit

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

WIN! KEYBOW MECHANICAL KEYBOARDS
The MagPi
The official Raspberry Pi magazine
Issue 80 | April 2019 | magpi.cc
THE 50 BEST RASPBERRY PI TIPS
HACK LEGO BOOST
Take control of Lego with Raspberry Pi
PLUS!
52 PAGES OF PROJECTS & GUIDES
ESSENTIAL HACKS, HINTS & TRICKS!
TOP 10 Raspberry Pi HATs

Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

Rolling Monthly Subscription

- **Low Monthly Cost** (from £5)
- **Cancel at any time**
- **Free delivery to your door**
- **Available worldwide**

Subscribe for 12 Months

£55 (UK)	£90 (USA)
£80 (EU)	£95 (Rest of World)

Free Raspberry Pi 3A+ with 12 Month upfront subscription only (no Raspberry Pi 3A+ with Rolling Monthly Subscription)

Subscribe online: magpi.cc/subscribe

LIMITED OFFER!

FREE!

Raspberry Pi 3A+

WITH YOUR 12 MONTH PRINT SUBSCRIPTION

WORTH £25

This is a limited offer. It replaces our usual offer of a free Pi Zero W.
Offer subject to change or withdrawal at any time.

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
WIN!
KEYBOARD & MOUSE KITS
DEVELOP AN ANDROID APP
MAKER SPECIAL!
MONTH OF MAKING
11 projects to create & share
ADD AI TO YOUR PROJECTS
SET UP A SMART CAR CAMERA
TOP 10 Home Automation Projects

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

THE 50 BEST RASPBERRY PI TIPS

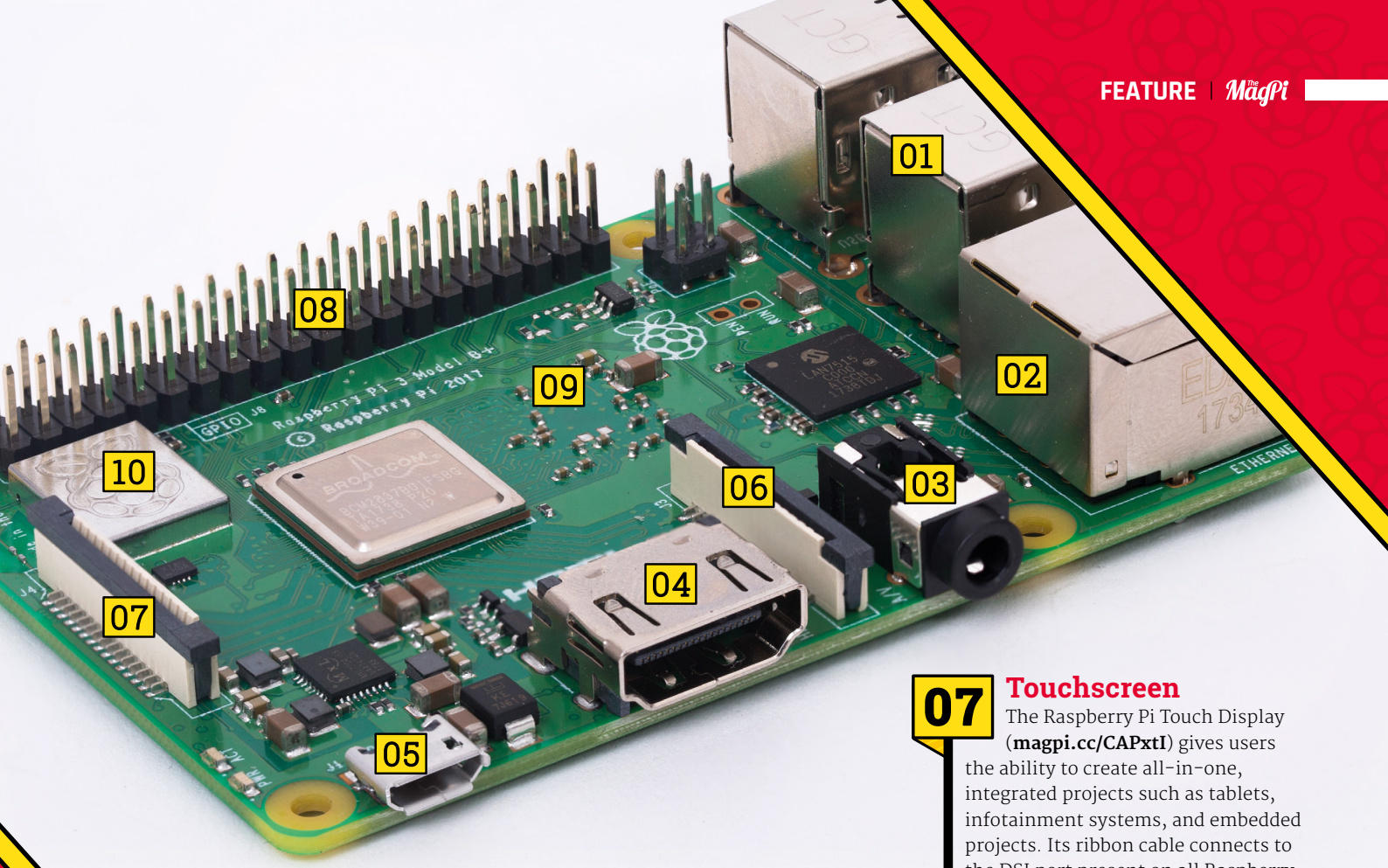
MASTER RASPBIAN & THE RASPBERRY PI WITH THESE HACKS, HINTS & TRICKS

01 Boot from USB

You can boot some Pi models (3B, 3B+, 3A+, 2B v1.2) from a USB port instead of using an SD card. It'll be faster and more reliable for long-term use, plus you get lots of extra storage space. See the official documentation at **magpi.cc/eXHfjq**.

02 Power over Ethernet

The official Raspberry Pi PoE HAT enables you to power the Raspberry Pi 3 Model B+ via an Ethernet cable. Great for those projects that require a constant wired connection.



03 Audio output

All Pi models have a composite out port for connecting to analogue devices. Pi B+ and later combine the audio out and composite out on to the same 3.5 mm jack plug. This requires a particular type of lead, with audio left on the tip, audio right on ring 1, ground on ring 2, and video on the sleeve. See **magpi.cc/tDfjLV**.

04 Switch sound options

The HDMI port outputs audio as well as video. If your display lacks a built-in speaker, you'll want to use the speaker jack. If so, right-click on the volume icon in the menu bar and switch from HDMI to Analog. Alternatively, use the command line:

```
amixer cset numid=3 1
```

Set the last number to '1' for Analog, or '2' for HDMI. Use raspi-config to change the audio setting at boot:

```
sudo raspi-config
```

Choose Advanced Options > Audio and 'Force 3.5 mm ('headphone') jack' or 'Force HDMI' to set this as the default.

05 Power it up

The power requirements of the Raspberry Pi increase as you make use of the various interfaces. The GPIO pins can draw 50 mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16 mA. The HDMI port uses 50 mA, the Camera Module requires 250 mA, and keyboards and mice can take as little as 100 mA or quite a bit more. Buy the Official Power Supply and check the power rating of your devices: **magpi.cc/xfszUR**.

06 Use a camera

You can use a webcam, but Raspberry Pi has a dedicated option: the Camera Module (**magpi.cc/jbKzbf**). This is attached to the CSI (Camera Serial Interface). The Camera Module is used to take high-definition video and still images. It's easy to use for beginners, but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries Raspberry Pi has developed to create visual effects.

07 Touchscreen

The Raspberry Pi Touch Display (**magpi.cc/CAPxtI**) gives users the ability to create all-in-one, integrated projects such as tablets, infotainment systems, and embedded projects. Its ribbon cable connects to the DSI port present on all Raspberry Pi boards (except Raspberry Pi Zero and Zero W).

08 GPIO pins guide

Open a Terminal and enter:

```
pinout
```

...for a quick visual guide to the GPIO pins. Or enter:

```
pinout -x
```

...to open the web browser and point it to the interactive model at **pinout.xyz**.

09 Hardware diagrams

You can see the mechanical drawings for the Raspberry Pi 3B+ (and all other models) on the Raspberry Pi website: **magpi.cc/WWFUji**.

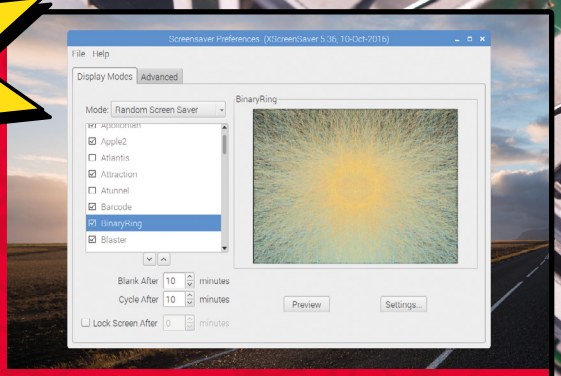
10 Wireless hotspot

Many Raspberry Pi models feature a wireless LAN adapter to connect to local network. But you can also use the access point to create a wireless hotspot. The process is a little long, but it's great for learning more about how wireless LAN works. Read more at **magpi.cc/nnriRT**.

RASPBIAN TIPS

Customise the Raspberry Pi operating system with these tips and tricks.

By Lucy Hattersley



▶ Add some visual flair to the Raspbian interface using the Screensaver program

11 Demo programs

Raspbian comes with a bunch of demo programs that demonstrate various features. Find them here:

```
cd /opt/vc/src/hello_pi
```

They're written in C. Compile them with:

```
./rebuild.sh
```

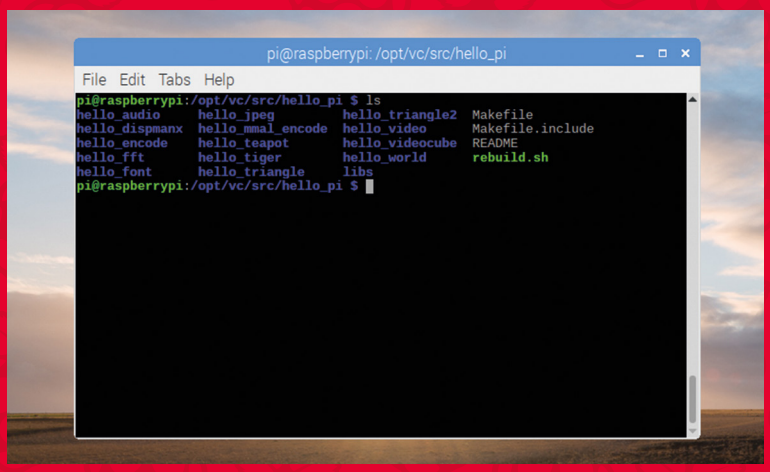
You will now be able to run the .bin files in each example folder. For instance, run:

```
cd hello_tiger
./hello_tiger.bin
```

...to see a large rotating image of a tiger on the screen. For more info, see **magpi.cc/tpNaGV**.

12 Screensaver

By default, the Raspberry Pi Desktop doesn't have easy-to-use screensaver software installed. Firstly, you should install the X Windows screensaver application.



▼ Raspbian has a selection of demo programs hidden in the **opt** directory

```
sudo apt-get update
sudo apt-get install xscreensaver
```

This may take a few minutes. The screensaver application will then appear under Preferences in the main desktop menu. This provides many options for setting up the screensaver, or disabling it.

13 Understand config.txt

Note: this is for advanced users only! The Pi uses a configuration file instead of the BIOS found on a conventional PC. This file is normally accessible as **/boot/config.txt** from Linux, and must be edited as root.

```
sudo nano /boot/config.txt
```

Any changes will only take effect after you have rebooted your Raspberry Pi. After Linux has booted, you can view the current active settings using the following commands:

```
vcgencmd get_config <config>
```

...displays a specific config value; for example, `vcgencmd get_config arm_freq`.

```
vcgencmd get_config int
```

...lists all the integer config options that are set (non-zero).

```
vcgencmd get_config str
```

...lists all the string config options that are set (non-null).

A range of options can be edited in the **config.txt** file – see **rpf.io/configtxt**.

SECURITY TIPS

When connecting your Pi to the outside world, you'll need to think about security

14 Free up space

You can save space on your SD card for projects, documents, and code by removing some of the programs provided in Raspbian by default. You can remove software from the command line:

```
sudo apt purge mathematica
sudo apt clean
sudo apt autoremove
```

But it's much better to choose Menu > Preferences > Recommended Software. Clear the tick next to the unwanted program and click OK.

15 Kernel update (& rollback)

If you use the standard Raspbian update/upgrade process, this will automatically update the kernel to the latest stable version. This is the recommended procedure. However, sometimes you may wish to update to the latest 'bleeding edge' or test kernel.

```
sudo rpi-update
```

The rpi-update utility will download the latest (unstable, testing) kernel version and copy all required files onto your system. Note that the latest kernel from rpi-update is not guaranteed to work correctly! Make sure that it doesn't conflict with your distribution packages. It doesn't provide a way of automatically uninstalling the files.

To revert to the current stock Raspbian kernel after trying rpi-update or a custom kernel, you need to reinstall both these packages, by running:

```
sudo apt-get install --reinstall
raspberrypi-bootloader raspberrypi-
kernel
```

16 Change your password

With the standard install of Raspbian, the default username is **pi** and the password is **raspberry**. If you haven't at least changed this password, anyone can log into your Pi! Go to Menu > Preferences > Raspberry Pi Configuration. In the System tab, click Change Password, enter one, confirm it, and click OK.

17 Keep Raspbian up to date

From time to time, security vulnerabilities are found in software, so it's always best to get the latest versions on a regular basis. In a Terminal, type `sudo apt-get update` to update your package lists, then `sudo apt-get dist-upgrade` to get the latest version of the packages on your system.

18 Install Fail2ban

If someone wants to hack into your Pi, they may try to guess your username and password. It will probably take a lot of attempts, but this is called 'brute-forcing'. To prevent this, you can install a program called Fail2ban. Install using `sudo apt install fail2ban` and a user will be banned for ten minutes if they fail login five times.

19 Change default SSH port

If you change the default SSH port, anyone trying to connect would need to know which port to use. To alter it, `sudo nano /etc/ssh/sshd_config` and change the line #Port 22 to read Port 2222 (without the #) or another memorable number. Save and exit (**CTRL+O**, **ENTER**, **CTRL+X**), then restart SSH with `sudo service ssh restart`.

20 Turn off interfaces you don't need

The safest way to stop hackers is to not have any network connection, but another way of shutting down the system a bit more is to go into Main Menu > Preferences > Raspberry Pi Configuration and select the Interfaces tab. Make sure everything you don't need is disabled.

CODING AND MAKING TIPS

Get more from your makes with Raspberry Pi.

By Lucy Hattersley

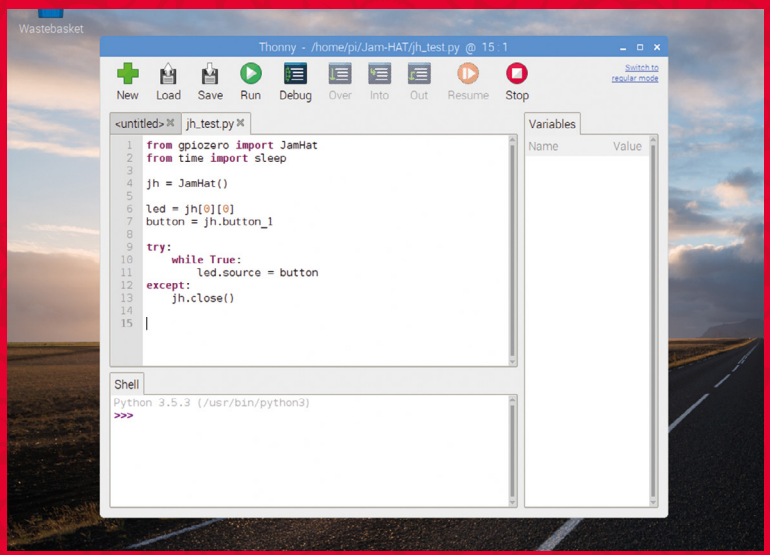
21 Switch to Thonny

If you're still programming in Python IDLE or using Text Editor, then switch to Thonny. It is just as easy as the other programs, but has a simple debugger that makes it much easier to fix flaws in your code. Press **CTRL+F5** instead of **F5** to run a program step-by-step. Unlike more advanced IDEs, you don't need to set breakpoints. Press **F6** for a big step and **F7** for a small step. It's ideal for makers because it enables you to fix problems, but isn't so involved that you get completely absorbed in code and so you can carry on with your build.

22 GPIO Zero

Make sure you learn to code using the GPIO Zero Python library (now in version 1.5). This boilerplate library is designed to take the hassle out of using electronics components and kits with the Raspberry Pi. With GPIO Zero you can spend more time putting your project together, and less time figuring out how to get it to work. Bookmark the documentation: **gpiozero.readthedocs.io**.

▼ The Thonny IDE (integrated development environment) makes it easier to debug your programs without getting too distracted



23 Make circuit diagrams

Fancy making a circuit board diagram like the ones you see in *The MagPi* magazine? The software you're looking for is Fritzing. It's currently in beta, and we mostly use the Windows version to create the images you see in the magazine. However, a version is available for Raspberry Pi.

```
sudo apt-get update
sudo apt-get install fritzing
```

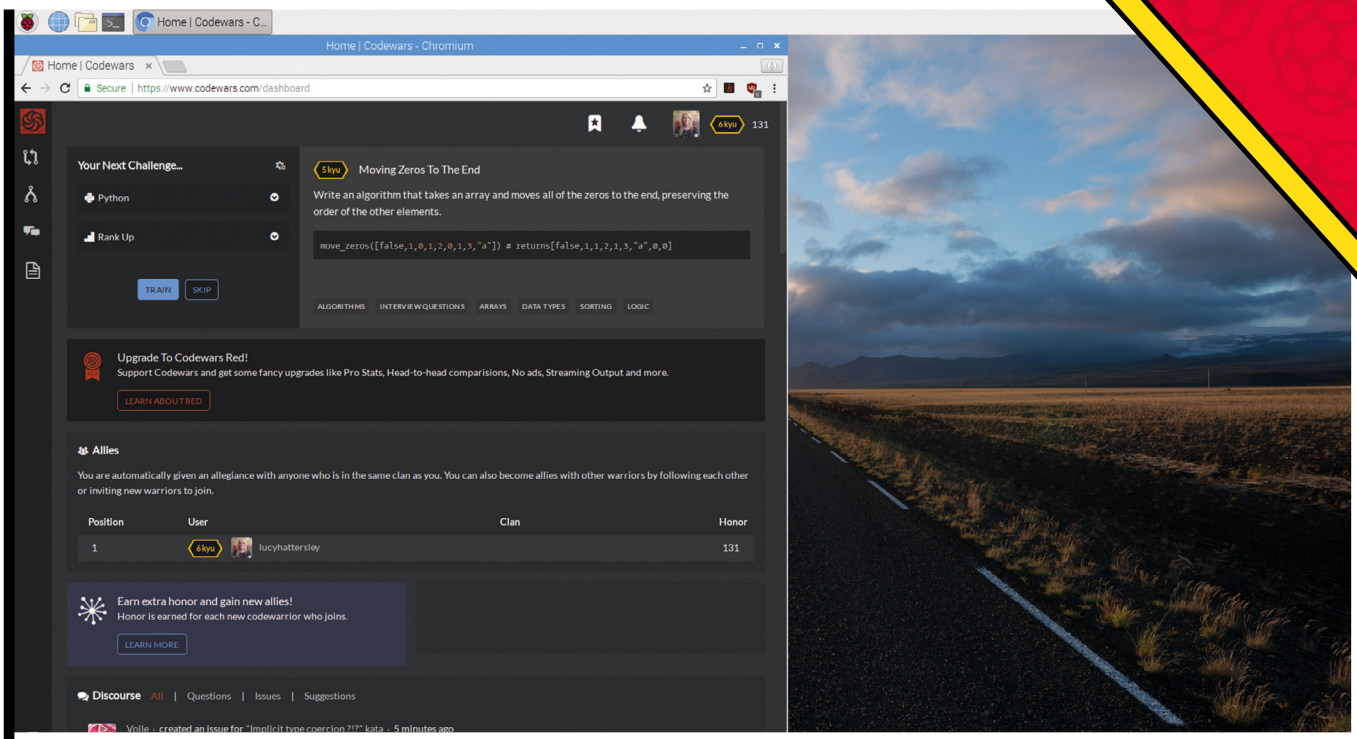
Restart your Raspberry Pi and you'll find the program under Menu > Programming. We are currently having trouble opening the parts bin on the Raspberry Pi; if any readers have a fix, then please let us know.

24 Raspberry Pi and Arduino

The Raspberry Pi works fantastically well with an Arduino attached. Despite looking visually similar, the two boards are very different. The Raspberry Pi is a complete single-board computer, while the Arduino range are microcontrollers. They work fantastically well together because the Arduino is handy for controlling parts, with the Raspberry Pi better for the higher-level interface and master control. You can install the software to control an Arduino from Raspberry Pi:

```
sudo apt-get upgrade
sudo apt-get install arduino
```

Arduino programs are called 'sketches', and are based on the C programming language. For more info, see our book on learning C (**magpi.cc/learn-c-book**) and our guide to Arduino and Pi in issue 67 (**magpi.cc/67**).



25 Codewars

One of the best ways to boost your programming skills is to complete challenges. If you are looking for some programming challenges, then sign up for Codewars on your Raspberry Pi. The website (**codewars.com**) has a video game style, and you get points and advance through levels by completing challenges (kata). We've even seen the latter being used in job interviews and to gain entry to coding boot camps.

26 Learn to solder

There's a technique to soldering wires together called the 'Western Union' or 'Lineman' splice, and you should learn it if you're going to solder (**magpi.cc/mckKVW**). It was developed during the introduction of the telegraph (hence the name), and the wrapping pattern is designed to tighten as the two ends pull against each other.

27 Make a maker kit

We can't recommend highly enough putting together a maker kit. This should contain a wide range of electronics parts, bits, and tools. Find a good components case and put in it a selection of resistors, capacitors, LEDs, buttons, potentiometers, a piezo buzzer, and a breadboard with lots of jumper cables. You can pick up kits and parts from most of the Raspberry Pi resellers, or visit the Raspberry Pi store in Cambridge. We use a half-size breadboard in our diagrams, so if you get that type you'll find it much easier to follow our projects: **magpi.cc/dXHByV**.

28 Second-hand

Get to know your local second-hand ('thrift') store. Used objects can be a great source of inspiration, and upcycling old tech with new Raspberry Pi parts is a popular pastime. High-profile maker Martin Mander has rescued old radios, televisions, and toys. Follow him on Twitter (**@MartinWMander**) for inspirational ideas.

29 Share early and often

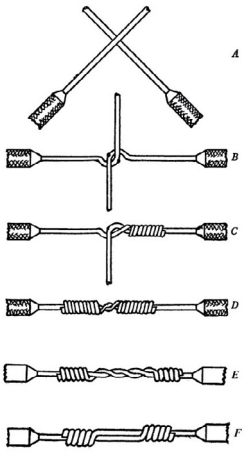
The best way to get inspiration and help for your project is to share it with the Raspberry Pi community. Last issue we ran #MonthOfMaking on Twitter, to encourage our readers to share projects with one another. But don't wait till next year – share your project on our forum (**magpi.cc/forum**), Facebook (**magpi.cc/facebook**), or Twitter (**@TheMagPi**). The more other readers see your project, the more likely you'll be to finish it.

30 Tips for makers

Element14 has two collections of Tips for Makers packed with interesting ideas. Sign up with the Element14 website to access them: **magpi.cc/XQaCyC**.

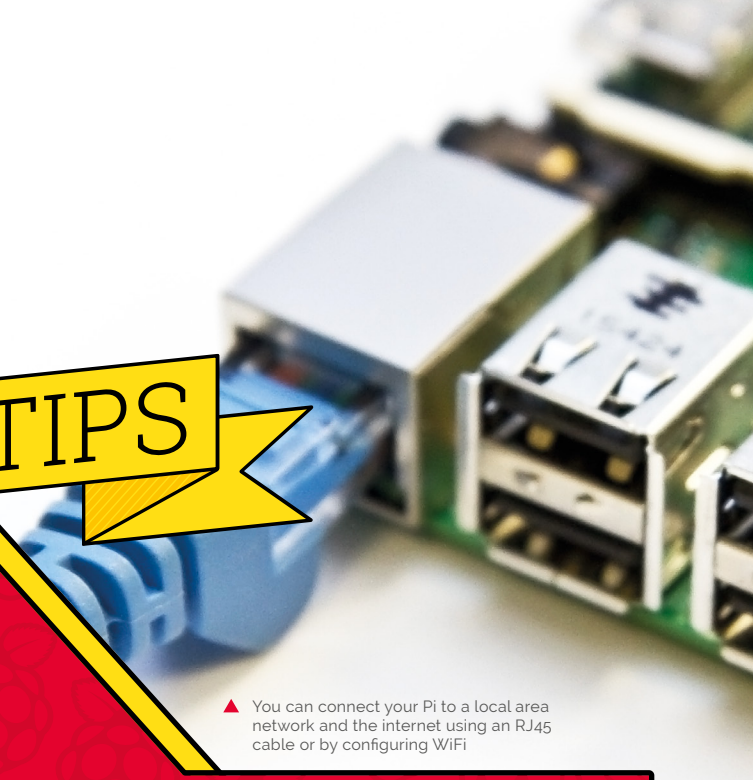
▲ Codewars is a good web resource for practising and flexing your programming muscles

▼ Learning the 'Western Union' splice technique will make your soldering joints stronger



NETWORKING TIPS

Need to know more about Raspberry Pi networking? Here are some of the basics and some interesting tips if you're already networked up



▲ You can connect your Pi to a local area network and the internet using an RJ45 cable or by configuring WiFi

You'll Need

- Network cable
- Internet router
- Internet connection

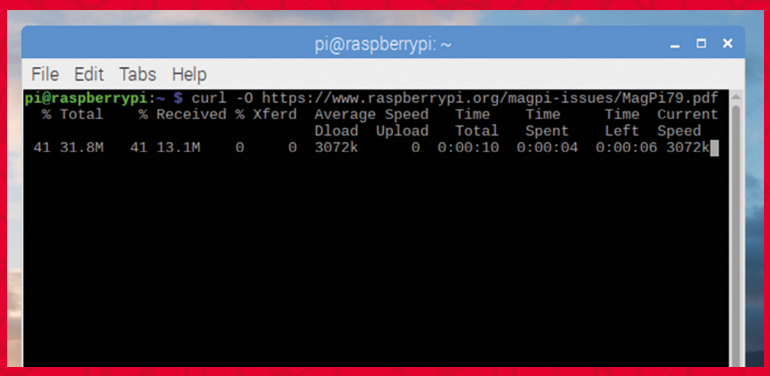
31 Wired or wireless

The Raspberry Pi has two standard ways of connecting to a network: with an Ethernet cable or on-board WiFi (on newer models). The Ethernet cable is usually either a Cat5e or Cat6 (for fast networks) type, with an RJ45 connector on each end. One end plugs into a router or hub and the other end into the Raspberry Pi. Wired connections are generally faster than WiFi and will connect as soon as the cable is plugged in. Home wireless network connections can usually be connected to by just selecting the network and typing in a password – see **magpi.cc/HiDkyk**.

32 Know your IP

An IP (Internet Protocol) address is a number that is allocated to your Raspberry Pi so that other computers know where to find it on the network. There are several ways to find your Pi's current IP address. In a Terminal window, there are a couple of commands you can use. The old command (which still works) is `ifconfig`. There is a new shorter command now, `ip`, which will provide similar information. Your local IP address will be the number that looks something like 192.168.0.34. You can also find your IP address by hovering over the network icon (top right) of your desktop.

▼ You can use the curl tool to download files, such as PDFs of *The MagPi* issues


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ curl -O https://www.raspberrypi.org/magpi-issues/MagPi79.pdf
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
41 31.8M 41 13.1M 0 0 3072k 0 0:00:10 0:00:04 0:00:06 3072k

33 IPv4 and IPv6

The terms IPv4 and IPv6 are descriptions of the internet protocol and address. IPv4 uses 32-bit numbers for the address and looks something like: 192.168.1.23. As the internet has now grown so large, the limited number of available IPv4 addresses has become a problem. A new version of IP (IPv6), using 128 bits for the IP address, has been defined. In July 2017, the final definition of the protocol was published, although IPv6 has been in use for over 20 years. An IPv6 address looks something like: fe80:94c:38c8:4600:7b33:64bb:baf7:e866. On a home network you will normally be using IPv4 format addresses, but look out for that to change in the future.

34 Useful commands

You can do quite a lot with network commands in the Terminal. Some commands need to be installed first. The ping command is there by default. If you type `ping` and then either an IP address or a web domain, it will tell you how long it takes to send a message to that computer/server and then get a reply. Cat is also pre-installed. Type `cat /sys/class/net/wlan0/operstate` to see the status of your WiFi connection. Install whois with `sudo apt-get install whois` and then type `whois` with a domain name and you will find the registration details of that domain.

35 Networking files

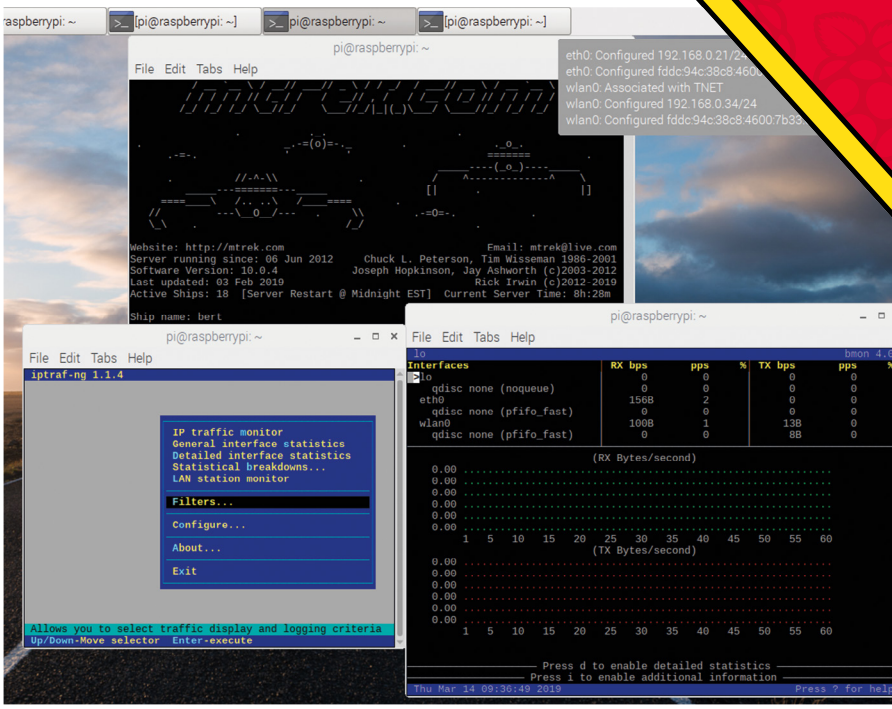
The Raspberry Pi has several files where it holds details about networking. It's quite useful to know where some of these are, as you may need to change some of the information for custom configurations. The file **/etc/hosts** has a list of names that map to IP addresses. You could have an entry that says '192.168.0.3 myserver' and that would mean that if you typed 'myserver' into a web browser address bar, the network would try to connect to 192.168.0.3. Other useful files are **/etc/networks**, which has network names mapped to IP addresses; **/etc/protocols**, which has a list of all the network protocols that can be used; and **/etc/services**, which has a list of all the network services available.

36 Network traffic

If you want to know a bit about what is happening between your Pi and the rest of the network, there are a few commands you can install. A simple stats command is available with `sudo apt-get install vnstat`. When you run this command, it will give you some basic data transmission totals. If you want some extra tools, you could install **iptraf-ng**, which has a menu of different tools and filters to analyse the network traffic. To see some real-time data transfer stats, install **bmon**, which shows data quantities per second.

37 Telnet

Although Telnet is a very old networking system (it was first designed in 1969), it is still available and can be installed on the Raspberry Pi using `sudo apt-get install telnet`. Telnet is a bit like a very early version of websites and when this is installed, your Pi can connect to any Telnet server. For example, enter `telnet towel.blinkenlights.nl 23` to see an ASCII version of *Star Wars*, or try `telnet mtrek.com 1701` for a *Star Trek* game. There are lots of Telnet sites still out there, some of which can be found at: **telnet.org/htm/places.htm**.



▲ You can use lots of different tools to find out about your Raspberry Pi network connections; you can even play online games in the Terminal window!

38 PiServer

PiServer is a useful tool for networking many Raspberry Pi boards together without needing SD cards in them. For info on how to set it up, go to **magpi.cc/RQDQXx**.

39 Downloading files with curl

You can download files directly in the Terminal with the curl tool. Use `curl -O <URL>`. See **magpi.cc/JrKoTA** for more info.

40 Headless wireless

If you use the Raspberry Pi in a 'headless' mode (without the Raspbian interface), it can be tricky to connect to a wireless network. You will need to define a **wpa_supplicant.conf** file for your particular wireless network and put this file in the **boot** folder. See here for more information: **magpi.cc/RHviuV**.

COMMAND LINE TRICKS

Looking to up your command line game? **PJ Evans** introduces the most popular shortcuts and tricks

41 Piping

Applications in Linux systems, such as Raspbian, have three ways of communicating with you and other applications: standard in (stdin), standard out (stdout), and standard error (stderr). Output from a command-line application arrives on stdout and you can use the pipe character '|' to send that output directly into another application, like this:

```
ls -l | wc -l
```

Here we get a long-form listing of the current directory and pipe that output into 'wc', a counting application. The `-l` argument means 'count the lines', so we now have a 'new' app that tells us how many items are in a directory. You can chain as many commands as you wish to achieve complex tasks.

42 Output redirection

Try this:

```
curl https://cataas.com/cat
```

The command downloads a cat image – but you're on the Terminal, which can't show graphics. The image arrives as a lot of nonsense on stdout. We can use redirection to save it as a file like this:

```
curl https://cataas.com/cat > cat.jpg
```

The '>' operator tells Raspbian to put everything arriving on stdout into the file instead of your screen. You can also use '>>' which appends to an existing file, or '<' to feed the contents of a file into a command.

43 Run multiple commands

You can put multiple commands together on a single line so that they execute one after the other:

```
command1; command2; command3
```

To only proceed if the previous command has succeeded:

```
command1 && command2 && command3
```

For example:

```
mkdir ~/myDir && cd ~/myDir && nano myFile.txt
```

...will create a directory, move into it, and open up the nano text editor with a new file, **myFile.txt**. You can save this line somewhere for future use or create an alias (see Tip 47).

44 Run background tasks

Running a command 'in the background' enables you to issue other commands while it works. If you're using the Desktop, the simplest solution is to open another Terminal window. Alternatively, to have a process run in the background, add an ampersand (&) at the end of the command:

```
very-long-command &
```

This works well, but any output to stdout and stderr will appear on your console screen. To avoid this, use 'nohup':

```
nohup very-long-command &
```

Now all output is written to **nohup.log** in your current directory.

45 Watching log files

Sometimes when a process is generating output, you might need to follow what's happening. For example, your web server isn't working quite right, so you can troubleshoot the problem by looking at the logs. To show the contents of any file on-screen:

```
cat text.txt
```

However, if the log file is huge, you'll struggle to cope. To show the file in 'pages':

```
less text.txt
```

To just see the last few lines:

```
tail text.txt
```

Best of all, to 'follow' the log file as it grows:

```
tail -f text.txt
```

46 A miscellany of shortcuts

Tilde – The tilde character '~' always refers to your home directory in paths. So `cd ~` always takes you home (as does `cd` on its own).

TAB autocompletion – Whenever possible, Raspbian will attempt to complete the command name you are typing. Try typing `whoa` followed by **TAB**. It will expand to `whoami`. This also works for paths.

History – Forgotten that command? Your Pi is keeping track. Pressing the up arrow key repeatedly goes through previous commands. You can also run `history` to get a full list.

!! – Two exclamation marks means 'the previous command'. Forgot to use sudo? Don't type the command again, just enter `sudo !!`.

47 Using aliases

If there's a long command you're going to be using a lot, create an alias for it. Aliases are kept in a hidden file (non-existent by default) called **.bash_aliases**, in the home directory. To create or view contents:

```
nano ~/.bash_aliases
```

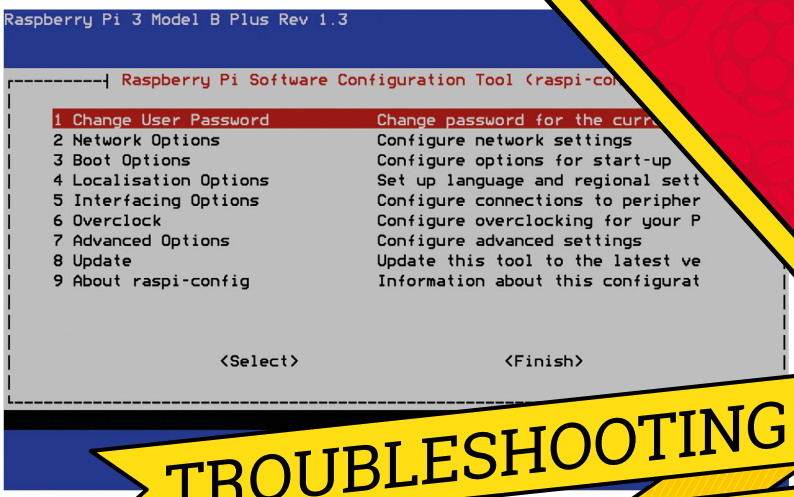
For each alias you want to create, enter a line following this pattern:

```
alias howmany="ls -l | wc -l"
```

Now you can run `howmany` for that command.

48 It's got stuck!

Stuck with a crashed program on your screen? To gain access back to the prompt, try **CTRL+C** to stop the process, or **CTRL+Z** to put it in the background. Now enter `killall <commandname>`. You can use `ps ax` to show a list of running processes. Still there? Try `sudo killall <commandname>`. As a last resort, use `sudo kill -9 <processid>` (get the process ID from `ps ax`).


Raspberry Pi 3 Model B Plus Rev 1.3
Raspberry Pi Software Configuration Tool (raspi-co
1 Change User Password
Change password for the curr
2 Network Options
Configure network settings
3 Boot Options
Configure options for start-up
4 Localisation Options
Set up language and regional sett
5 Interfacing Options
Configure connections to peripher
6 Overclock
Configure overclocking for your P
7 Advanced Options
Configure advanced settings
8 Update
Update this tool to the latest ve
9 About raspi-config
Information about this configurat
<Select>
<Finish>

TROUBLESHOOTING TIPS

▲ Some command-line applications take on a basic graphical appearance to make them easier to use. The most likely you are to see is raspi-config

49 Find out more

Many commands come with a huge array of 'arguments' that control behaviour. These are in the form of 'switches' (simple flags) or 'parameters' that provide additional info such as an input file. Most commands have an accompanying 'man' page (short for manual). To see if a 'manpage' exists for your command, just enter:

```
man command
```

You can even learn about 'man' itself:

```
man man
```

Alternatively, many commands respond to the switch `--help` (or `-h` for short) to provide a summary of options.

50 Get in remotely

We've all been there: happily working away in Raspberry Pi Desktop and something goes awry. Now the screen has locked up, your Doom high score is lost, and the only option is to unplug the Pi, which risks data corruption. Well, you may be able to get in using SSH, the 'Secure SHell' which allows you to log into your Pi (so long as you've already enabled SSH on it) from another computer and get to the command line:

```
ssh raspberrypi.local
```

Then, if you get to a prompt:

```
sudo shutdown -h now
```

...and your Pi is rescued.

THE Official

RASPBERRY PI PROJECTS BOOK

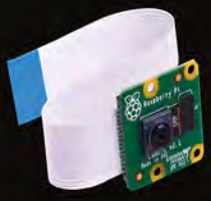
VOLUME 4

200 PAGES OF IDEAS & INSPIRATION

DIY Games Console

Build a Robot

Make a Magic Mirror



Set up a Spy Cam

55 PROJECTS & GUIDES

FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4

200 pages of ideas & inspiration

£12.99
200 pages of
Raspberry Pi

THE Official
RASPBERRY PI PROJECTS BOOK
VOLUME 4

Amazing hacking and making projects
from the makers of The MagPi magazine

Inside:

- How to get involved with the Pi community
- The most inspirational community projects
- Essential tutorials, guides, and ideas
- Expert reviews and buying advice

Available now magpi.cc/store

plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

Part 01

Hack Lego Boost with Raspberry Pi

The Lego Boost robotics kit is designed to make learning to build and program robots fun and easy. Let's hack it with some Python

MAKER

Mike Cook

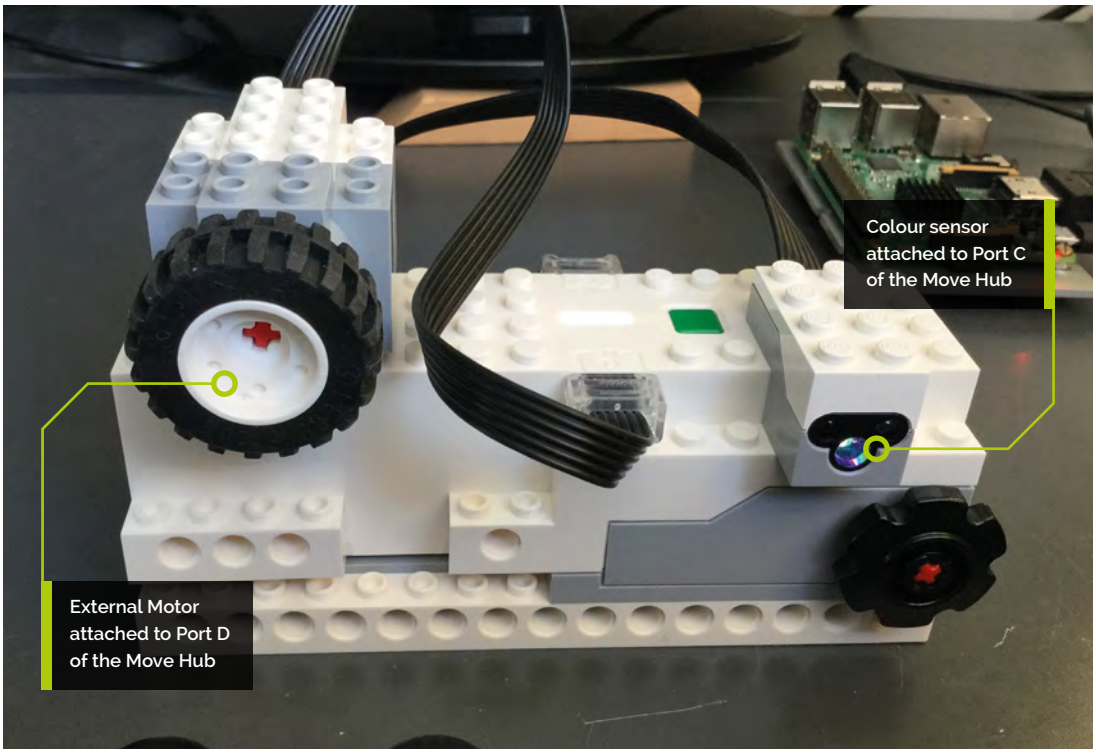
Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/TPaUfT

The Lego Boost is designed to be run from an app on an Android or iPad tablet, using a graphics programming language not unlike Scratch. It makes a good job of this and is easy for kids to pick up, but these sorts of languages have their limitations. They can be inflexible and difficult to read, especially as the code gets bigger. By using Python, much more complex programs can created, many that are not possible with graphics-based code. So, give your Boost a boost by letting Python do the controlling.

01 What is Lego's Boost?

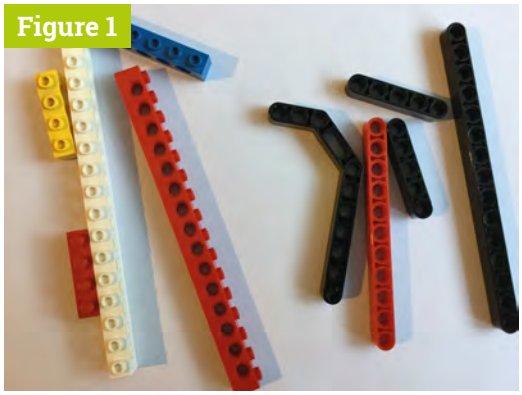
Unlike the previous Mindstorms robotics systems from Lego, the Boost system has no controlling brick to run code – instead, instructions representing the program are sent one at a time directly, over Bluetooth, in real-time to the Move Hub. This has built-in motors, LEDs, tilt sensor, and push-button. You can also plug into the Hub a smaller motor and a distance sensor or colour sensor. Also, Mindstorms uses Lego Technic beams for most constructions, known as a studless system, whereas the Boost uses the more conventional, studful brick form of construction. See **Figure 1**.



You'll Need

- Lego 17101 Boost Creative Toolbox **magpi.cc/JtUiDe**
- Raspberry Pi 3/3B+/3A+/Zero W

Figure 1



▲ **Figure 1** Studful bricks on the left, studless beams on the right

02 Lego's software

The standard Lego software provides a good structured learning system where the user builds a bit, programs it, and then moves on to building more. The big project is Vernie the robot and we'd recommend you do this first because the parts are already bagged up to make construction of this project easy. Vernie is built up in stages and the code you are asked to run at each stage gets increasingly complex. You can't progress to the next level without trying, or pretending to try, the current phase. We recommend you try this first, not least for the firmware updates it can offer to your Hub – **see Figure 2**.

03 Libraries

There are a few libraries for connecting Boost to Python – see a summary at **magpi.cc/iNfUQg** from Jorge Pe, author of the original library. He hacked the system by using a Bluetooth protocol sniffer. These days, Lego has released the protocol documentation at **magpi.cc/YfiAbL**, but it's a heavy read. We tried a few libraries and it seems the most developed and actively supported one is pylgbst

Figure 2

▲ **Figure 2** The Lego Boost program environment running on an iPad or Android

mike's_demo.py

DOWNLOAD THE FULL CODE:
magpi.cc/dhaAam

> Language: **Python**

```
001.  # #!/usr/bin/env python3
002.  # coding=utf-8
003.  # Mike's Demo - put the LEGO Boost trough its paces
004.  # By Andrey Pokhilko & Mike Cook Feb 2019
005.
006.  from time import sleep
007.  import time
008.  from pylgbst import *
009.  from pylgbst.movehub import MoveHub, COLORS
010.  from pylgbst.peripherals import EncodedMotor, TiltSensor,
      Amperage, Voltage
011.
012.  def main():
013.    print("Mike's Demo - put the LEGO Boost trough its paces")
014.    conn=get_connection_auto()
015.    try:
016.      movehub = MoveHub(conn)
017.      demo_voltage(movehub)
018.      demo_button(movehub)
019.      demo_led_colors(movehub)
020.      demo_motors_timed(movehub)
021.      demo_motors_angled(movehub)
022.      demo_port_cd_motor(movehub)
023.      demo_tilt_sensor_simple(movehub)
024.      demo_tilt_sensor_precise(movehub)
025.      demo_color_sensor(movehub)
026.      demo_motor_sensors(movehub)
027.      sleep(1)
028.      print("That's all folks")
029.    finally:
030.      conn.disconnect()
031.
032.  def demo_voltage(movehub):
033.      print("Reading voltage & current for a short time")
034.      def callback1(value):
035.          print("Amperage: %.3f" % value)
036.
037.      def callback2(value):
038.          print("Voltage: %.3f" % value)
039.
040.      movehub.amperage.subscribe(callback1, mode=Amperage.MODE1,
      granularity=0)
041.      movehub.amperage.subscribe(callback1, mode=Amperage.MODE1,
      granularity=1)
042.
043.      movehub.voltage.subscribe(callback2, mode=Voltage.MODE1,
      granularity=0)
044.      movehub.voltage.subscribe(callback2, mode=Voltage.MODE1,
      granularity=1)
045.      sleep(5)
046.      movehub.amperage.unsubscribe(callback1)
047.      movehub.voltage.unsubscribe(callback2)
048.
049.  def demo_button(movehub):
050.      global notPressed
051.      print("Please press the green button")
```

Top Tip

Colour sensor testing

The colour sensor only works when the object is between about 3 and 5 mm. Use tiles like part 4566179 and 4560181 to test the colours.

from Andrey Pokhilko. In fact, during the course of writing this, a new version was released. It is still not finished but, with a little tweaking, it is good.

04 Installing the pylgbst library

First off, we need to install some dependences. Open a Terminal window and enter:

```
sudo apt-get install python3-pip
sudo pip3 install pexpect
sudo pip3 install pygatt
```

Next, get the library:

```
wget https://github.com/undera/pylgbst/archive/master.zip
unzip master.zip
cd pylgbst-master
sudo python3 setup.py install
```

05 Fixing the colours

There seems to be an error in one of the files defining the colours for the RGB LED. Fix this by opening a Terminal and entering:

```
sudo nano /usr/local/lib/python3.5/dist-packages/pylgbst/constants.py
```

Scroll down using the cursor keys until you see:

```
COLOR_ORANGE = 0x09
```

…and change it to:

```
COLOR_ORANGE = 0x08
```

To save the changes, press **CTRL+X**, then **Y**, and **ENTER**.

06 Documentation

The bulk of the documentation is in the **README.md** file and we found that not all of it works with the Raspberry Pi. But, it does contain some short single-function examples. Basically, to get the Move Hub to do things, you send it a command, as you might expect; however, to get information back from the Move Hub, you have to subscribe to the appropriate stream. When you do, you specify the name of a function that will be called when there is new data from whatever sensor you subscribed from. You must unsubscribe from the sensor before your Python program finishes.

07 Demo example

In the library's **examples** folder is a **demo.py** program; unfortunately, this will not run without error as is – it seems to be more designed for fault finding on the Bluetooth connections than showing things working. We have rewritten this to add extra functions and correct the syntax errors – see the **mike's_demo.py** listing. While it will run with any Move Hub, incorporated into a model or not, we found it best to use a basic setup that didn't run away off the desk when the motors moved. Also, adding wheels to the motors made it easy to turn them when testing the motor read angle function.

08 Preparing the Hub

Refer to **Figures 3** to **5** with these instructions. Get the Move Hub and fix on top of it the sensor (part 6182145) and the motor (part 6181852) and plug them into ports C and D. It doesn't matter which port you use, the software will find them. Next, put a red axle stub (part 4142865) in each of the two Hub motors, and the external motor. Fit a wheel (part 4662228) on each of the Move Hub's axles, and a wheel and tyre (parts 6092256 and 4619323) onto the external motor. Finally, add two long beams (part 4508661) under the hub to stop the wheels from moving it.

Figure 3 The top of the minimum demo build

Figure 4 The front of the minimum demo build

Figure 5 The end of the minimum demo build

Figure 3

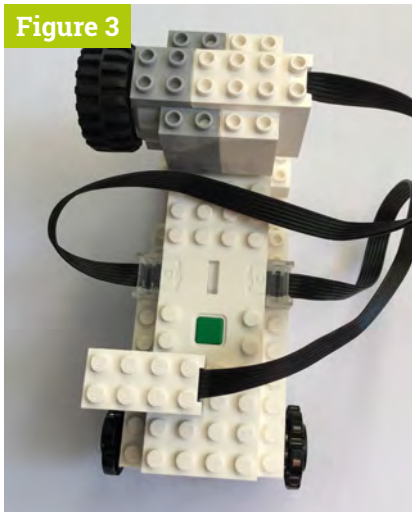


Figure 4

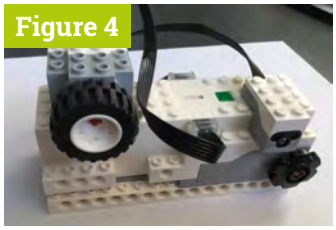
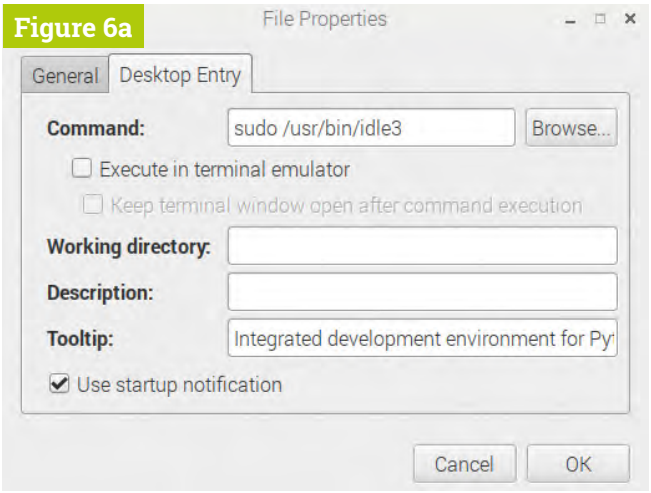
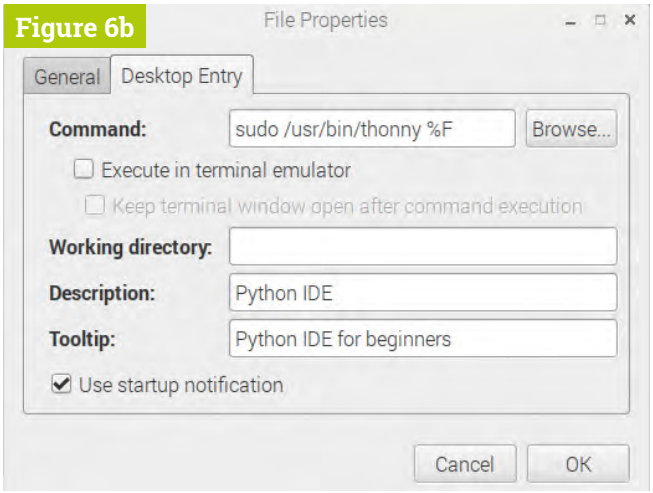


Figure 5

mike's_demo.py (continued)

```
052.      notPressed = True
053.
054.      def call_button(is_pressed):
055.          global notPressed
056.          if is_pressed :
057.             print("Thank you button pressed")
058.          else:
059.            print("Now it is released")
060.            notPressed = False
061.
062.      movehub.button.subscribe(call_button)
063.      while notPressed:
064.          sleep(0.4)
065.      sleep(0.4)
066.      movehub.button.unsubscribe(call_button)
067.
068.  def demo_led_colors(movehub):
069.      # LED colors demo
070.      print("LED colours demo")
071.      for colour in range(1,11):
072.          print("Setting LED colour to: %s" %
      COLORS[colour])
073.          movehub.led.set_color(colour)
074.          sleep(1)
075.
076.  def demo_motors_timed(movehub):
077.      print("Motors movement demo: timed")
078.      for level in range(0, 101, 10):
079.          levels = level / 100.0
080.          print(" Speed level: %s" %  levels)
081.          movehub.motor_A.timed(0.2, levels)
082.          movehub.motor_B.timed(0.2, -levels)
083.      print("now moveing both motors with one command")
084.      movehub.motor_AB.timed(1.5, -0.2, 0.2)
085.      movehub.motor_AB.timed(0.5, 1)
086.      movehub.motor_AB.timed(0.5, -1)
087.
088.
089.  def demo_motors_angled(movehub):
090.      print("Motors movement demo: angled")
091.      for angle in range(0, 361, 90):
092.          print("Angle: %s" % angle)
093.          movehub.motor_B.angled(angle, 1)
094.          sleep(1)
095.          movehub.motor_B.angled(angle, -1)
096.          sleep(1)
097.
098.      movehub.motor_AB.angled(360, 1, -1)
099.      sleep(1)
100.      movehub.motor_AB.angled(360, -1, 1)
101.      sleep(1)
102.
103.
104.  def demo_port_cd_motor(movehub): # Move motor on port
      C or D
105.      print("Move external motor on Port C or D 45
      degrees left & right")
106.      motor = None
107.      if isinstance(movehub.port_D, EncodedMotor):
108.          print("Rotation motor is on port D")
109.          motor = movehub.port_D
110.      elif isinstance(movehub.port_C, EncodedMotor):
111.          print("Rotation motor is on port C")
112.          motor = movehub.port_C
113.      else:
114.          print("Motor not found on ports C or D")
115.      if motor:
116.          print("Left")
117.          motor.angled(45, 0.3)
118.          sleep(3)
119.          motor.angled(45, -0.3)
120.          sleep(1)
121.
122.          print("Right")
123.          motor.angled(45, -0.1)
124.          sleep(2)
125.          motor.angled(45, 0.1)
126.          sleep(1)
127.
128.  def demo_tilt_sensor_simple(movehub):
129.      print("Tilt sensor simple test. Turn Hub in
      different ways.")
130.      demo_tilt_sensor_simple.cnt = 0
131.      limit = 10 # number of times to take a reading
132.
133.      def callback(state):
134.          demo_tilt_sensor_simple.cnt += 1
135.          print("Tilt # %s of %s: %s=%s" % (
      demo_tilt_sensor_simple.cnt, limit, TiltSensor.TRI_
      STATES[state], state))
136.
137.      movehub.tilt_sensor.subscribe(callback,
      mode=TiltSensor.MODE_3AXIS_SIMPLE)
138.      while demo_tilt_sensor_simple.cnt < limit:
139.          sleep(1)
140.      movehub.tilt_sensor.unsubscribe(callback)
141.
142.  def demo_tilt_sensor_precise(movehub):
143.      print("Tilt sensor precise test. Turn device in
      different ways.")
144.      demo_tilt_sensor_simple.cnt = 0
145.      limit = 50
146.
147.      def callbackTilt(roll, pitch, yaw):
148.          demo_tilt_sensor_simple.cnt += 1
149.          print(
      "Tilt #%s of %s: roll:%s pitch:%s yaw:%s" % (demo_
      tilt_sensor_simple.cnt, limit, roll,pitch,yaw))
150.
151.      # granularity = 3 - only fire callback function
      when results change by 3 or more
152.      movehub.tilt_sensor.subscribe(callbackTilt,
```





▲ **Figure 6** Changing the command so that IDLE or Thonny will run in supervisor mode

09 Running Mike's demo

Make sure Bluetooth is turned on by clicking on the Bluetooth icon on the top menu bar, run the Python code, and immediately push the green button on the Move Hub. The LED next to this will start flashing to indicate that the Bluetooth Hub is looking for something to pair with. This will turn a steady blue when connected and the demo will then start, printing out instructions on the screen. After the battery's voltage and current have been read out, the program will ask you to press the green button; you need to do this for the program to advance.

10 More demo instructions

After the green button is pressed, the distance sensor is tested; this repeats for 100 measurements. You can change the distance sensor reading by waving your hand in front of it: it returns a value in inches, to the nearest inch, for distances above one inch; for smaller distances, it returns finer values. The program converts this into millimetres for display. The tilt sensor tests require you to pick up and turn the Hub, and the motor angle read requires you to turn the motor wheels by hand. When no movement of the wheels has been detected over the last five seconds, the demo ends.

11 Further examples

The library's **examples** folder also contains other examples. Unfortunately, they don't work on the Raspberry Pi due to the method used to include packages. The `from . import *` line, which means from the current directory import everything, returns an error message saying the parent directory is missing. This might have something to do with the author not using a Raspberry Pi in his testing. Note, there is another problem but this is down to Lego, as it also happens on the tablets running the official software. Colours set or reported as cyan are actually green, so we have to cope with this in our code.

12 Move Hub connection problems

Here at the Bakery we have two Raspberry Pi boards: one with the current Raspbian release, and the other the previous one. Oddly enough, we found that running normally was fine with the older release, but with the newer one the code needed to be run in supervisor mode in order for it to connect. This can be done by typing `sudo IDLE3` in a Terminal window. If you want to permanently run IDLE in the supervisor mode from the desktop, go to the IDLE 3 menu entry, and right-click. Then choose the Properties menu, and select the Desktop Entry tab. Now edit the command box to put 'sudo ' in front of what is there. The same applies for Thonny – see **Figure 6**.

In conclusion

Now we've got the basics going, we're all ready to do some interesting stuff. Next month, we'll make a game showing how we can use our Lego Boost with the Pygame framework.

Top Tip

Lego part numbers

On the back of the cardboard play mat is a picture and part number for all parts in the Boost set.

Disclaimer

LEGO® is a trademark of the LEGO Group of companies, which does not sponsor, authorise, or endorse this article.

mike's_demo.py (continued)

```
    mode=TiltSensor.MODE_3AXIS_FULL, granularity=3)
153.     while demo_tilt_sensor_simple.cnt < limit:
154.         sleep(1)
155.     movehub.tilt_sensor.unsubscribe(callbackTilt)
156.
157. def callback_color(color, distance=None): # returns
    to nearest for distances > 1 inch
158.     global limit
159.     demo_color_sensor.cnt += 1
160.     correctColor = color
161.     if color == 5: # to correct for error in sensor
162.         correctColor = 6
163.     if distance != None:
164.        metric = distance * 2.45  # distance in mm
165.     else:
166.       metric = 0
167.     print('#%s/%s: Colour %s, distance %.2fmm' % (
    demo_color_sensor.cnt, limit, COLORS[correctColor],
    metric))
168.
169. def demo_color_sensor(movehub):
170.     global limit
171.     print("Colour & distance sensor test: wave your
    hand in front of it")
172.     demo_color_sensor.cnt = 0
173.     limit = 100 # number of times to take a reading
174.     try:
175.         movehub.color_distance_sensor.
    subscribe(callback_color)
176.     except:
177.        print("No colour & distance sensor found")
178.        sleep(2)
179.        return
180.     while demo_color_sensor.cnt < limit:
181.         sleep(0.5)
182.     # sometimes gives a warning - not sure why
183.     movehub.color_distance_sensor.unsubscribe(
    callback_color)
184.
185. def demo_motor_sensors(movehub):
186.     global resetTimeout, posA, posB, posE
187.     testTime = 5
188.     print("Motor rotation sensors test, move by hand
    any motor")
189.     print("Test ends after %s seconds with no change"
    % testTime)
190.     print()
191.     demo_motor_sensors.states =
    {movehub.motor_A: None, movehub.motor_B: None}
192.     resetTimeout = False ; external = False
193.     posA = 0 ; posB = 0 ; posE = 0
194.
195.     def callback_a(param1):
196.         global resetTimeout,posA
197.         last_a = demo_motor_sensors.states[movehub.motor_A]
198.         if last_a != param1:
199.             resetTimeout = True
200.         demo_motor_sensors.states[movehub.motor_A] = param1
201.         posA = param1
202.
203.     def callback_b(param1):
204.         global resetTimeout,posB
205.         last_b = demo_motor_sensors.states[
    movehub.motor_B]
206.         if last_b != param1:
207.             resetTimeout = True
208.         demo_motor_sensors.states[movehub.motor_B] = param1
209.         posB = param1
210.
211.     def callback_e(param1):
212.         global resetTimeout,posE
213.         last_e = demo_motor_sensors.states[
    movehub.motor_external]
214.         if last_e != param1:
215.             resetTimeout = True
216.         demo_motor_sensors.states[
    movehub.motor_external] = param1
217.         posE = param1
218.
219.     movehub.motor_A.subscribe(callback_a)
220.     movehub.motor_B.subscribe(callback_b)
221.     external = False
222.
223.     if movehub.motor_external is not None:
224.         demo_motor_sensors.states[
    movehub.motor_external] = None
225.         external = True
226.         movehub.motor_external.subscribe(callback_e)
227.
228.     timeOut = time.time() + testTime
229.     while timeOut > time.time():
230.       if resetTimeout :
231.          if external :
232.             print(
    "Motor A position %s \t Motor B position %s \t
    External Motor position %s" % (posA, posB, posE))
233.          else:
234.             print("Motor A position %s \t Motor B
    position %s " % (posA, posB))
235.          timeOut = time.time() + testTime
236.          resetTimeout = False
237.
238.     movehub.motor_A.unsubscribe(callback_a)
239.     movehub.motor_B.unsubscribe(callback_b)
240.
241.     if movehub.motor_external is not None:
242.         demo_motor_sensors.states[
    movehub.motor_external] = None
243.         movehub.motor_external.unsubscribe(callback_e)
244.
245. if __name__ == '__main__':
246.     main()
```

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi* magazine. She enjoys hacking and fixing technology, and building quirky projects. She's done the Samba all weekend long!

magpi.cc

Build a Samba file server

Use Samba/CIFS to share a Raspberry Pi directory with other computers on a network

It's easy to use a Raspberry Pi as a Samba file server where you can store backups and share files from all the other computers on your network.

Samba is the Linux implementation of the SMB/CIFS file-sharing standard used by Windows PCs and Apple computers, and widely supported by media streamers, games consoles, and mobile apps.

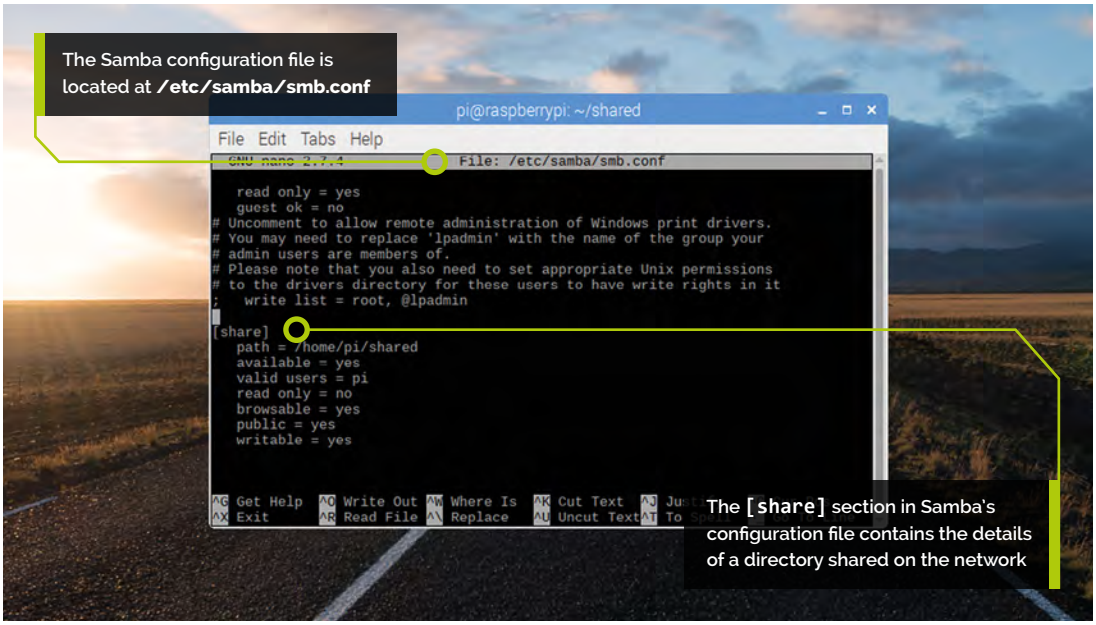
With Samba activated, you can quickly copy files from a computer on your network to a Raspberry using wireless LAN (or a direct Ethernet connection).

This tutorial assumes that you'll use a keyboard, mouse, and monitor to set up your file server; alternatively, you can enable SSH (**magpi.cc/ssh**) and connect to it remotely from another computer on your local network.

We also assume you're using a 32GB (or smaller) microSD card, which provides a reasonable amount of storage space without requiring any extra steps to make it accessible. However, if you need extra storage, it's possible to mount a large external USB drive and create a Samba entry for it.

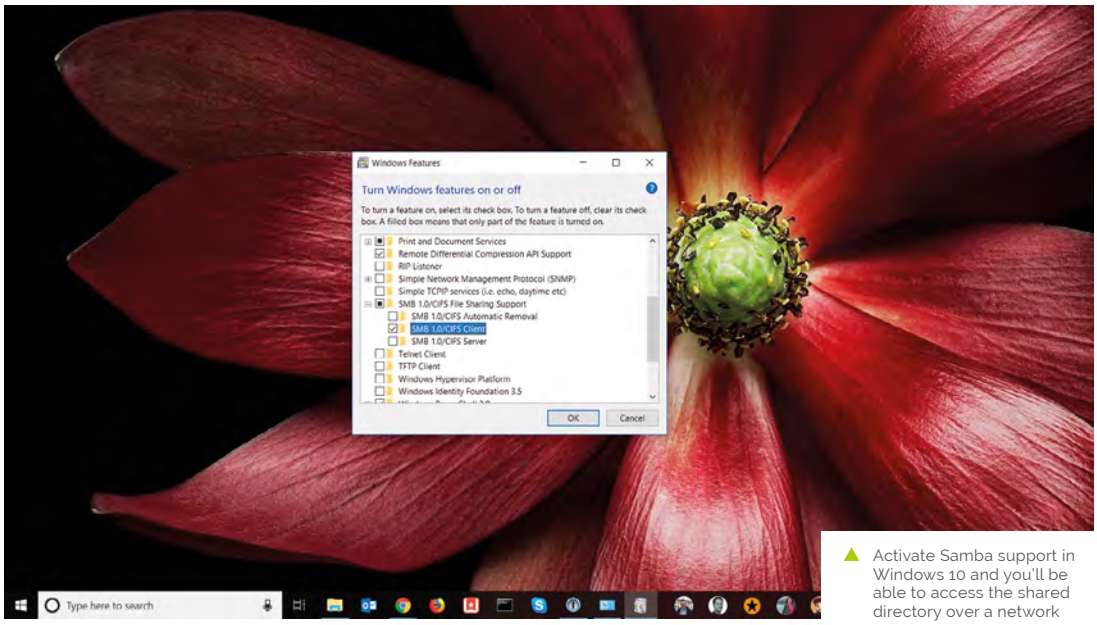
Alternatively, if you want to keep things compact, you can install Raspbian on microSD cards of up to 256GB, although we think it's wiser to install Samba on an external drive and use a smaller microSD to boot Raspbian and connect to it.

We suggest checking non-working SD cards online (**magpi.cc/2q97aGO**) before you buy, to make sure you get one that's fully compatible with the Raspberry Pi.


The Samba configuration file is located at /etc/samba/smb.conf
pi@raspberrypi: ~/shared
File Edit Tabs Help
File: /etc/samba/smb.conf
read only = yes
guest ok = no
Uncomment to allow remote administration of Windows print drivers.
You may need to replace 'lpadmin' with the name of the group your
admin users are members of.
Please note that you also need to set appropriate Unix permissions
to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin
[share]
path = /home/pi/shared
available = yes
valid users = pi
read only = no
browsable = yes
public = yes
writable = yes
^G Get Help ^O Write Out ^W Where Is ^K Cut Text
^X Exit ^R Read File ^\ Replace ^U Uncut Text
The [share] section in Samba's configuration file contains the details of a directory shared on the network

You'll Need

- Raspberry Pi
- microSD card
- Keyboard and mouse for setup
- Network and router (Ethernet recommended)


Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
Print and Document Services
Remote Differential Compression API Support
RIP Listener
Simple Network Management Protocol (SNMP)
Simple TCPIP services (i.e. echo, daytime etc)
SMB 1.0/CIFS File Sharing Support
SMB 1.0/CIFS Automatic Removal
SMB 1.0/CIFS Client
SMB 1.0/CIFS Server
Telnet Client
TFTP Client
Windows Hypervisor Platform
Windows Identity Foundation 3.5
OK
Cancel
Type here to search

▲ Activate Samba support in Windows 10 and you'll be able to access the shared directory over a network

Top Tip

Status report

You can see the status of the Samba drive using `sudo service smdb status`. And you can use `smbd restart` to turn the service off and on again.

Once set up, you can mount your home file server on all the other computers on your network, and use it as a convenient place to store everything from music files you want to share with your housemates, to backups of important documents and save-game files you'd like to share between computers.

We recommend using a wired Ethernet connection for stability and fast transfer speeds. The project will still work if you connect your Raspberry Pi via WiFi, although performance will be affected, particularly when it comes to copying over large files.

01 Install Raspbian

Download the latest version of NOOBS (or the latest Raspbian Stretch with Desktop image file) and install it to a microSD card. See **magpi.cc/quickstart** if you're unsure on how to set up a Raspberry Pi for the first time. Insert the microSD card into your Raspberry Pi and connect the keyboard, mouse, and HDMI cable. We suggest you connect an Ethernet cable between your Raspberry Pi and modem/router (otherwise use the wireless LAN icon to connect to a wireless network when booted up). Attach the power cable to boot up the Raspberry Pi.

02 Install Samba

Samba is available in Raspbian's standard software repositories. We're going to update our repository index, make sure our operating system is fully updated, and install Samba using APT. Open a Terminal and type:

```
sudo apt update
sudo apt upgrade
sudo apt install samba samba-common-bin
```

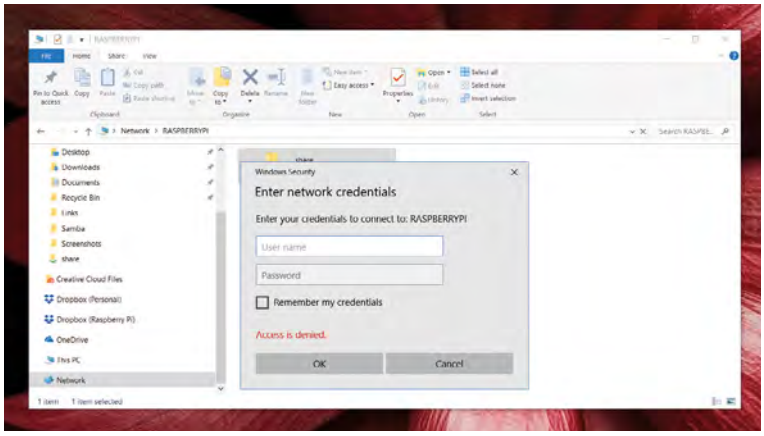
Answer **Y** when prompted.

03 Create shared directory

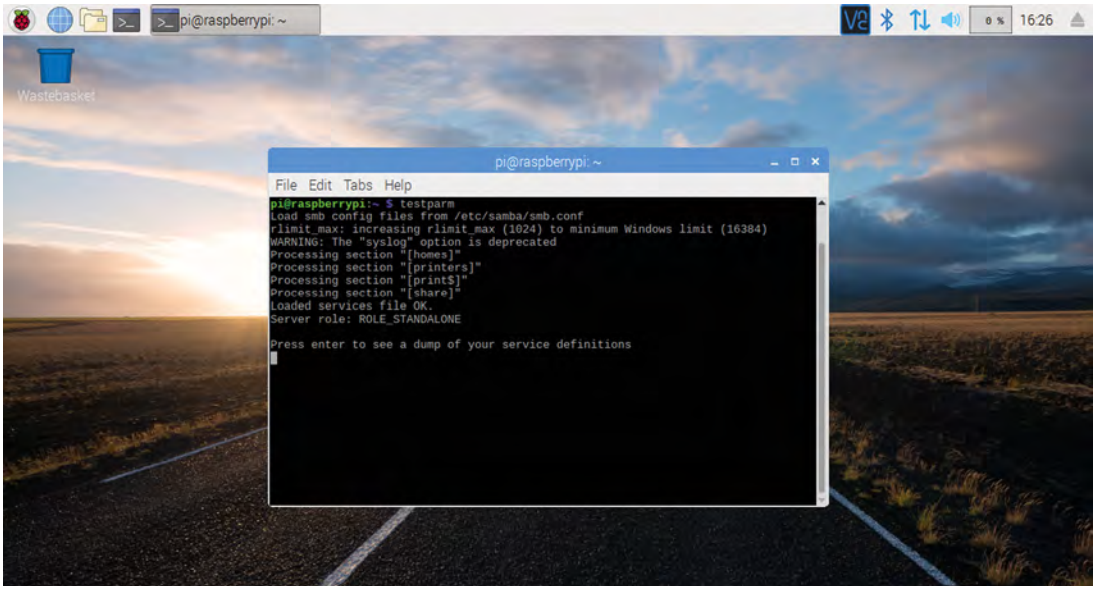
Now we're going to create a dedicated shared directory in our home directory.

```
sudo mkdir /home/pi/shared
```

You can put it anywhere, but ours is in the home directory (it makes it easier to access without adjusting permissions). You'll be able to access this



▼ Enter the credentials (username and password) created during the setup process to access the Samba share



Use testparm to test the edits you make to **smb.conf**

directory from other computers on your network. Files you drop in here are accessible on other computers, and files you drop into the directory from other computers will appear on your Pi.

04 Configure Samba

Now we need to edit Samba's configuration file to make the shared directory visible to other computers on the network. First, let's make a backup just in case you want to look at the original.

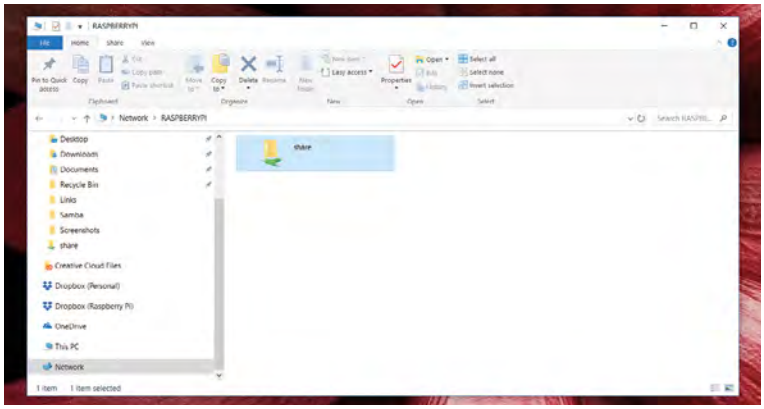
```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf_backup
```

Now let's edit the Samba configuration file. Enter:

```
sudo nano /etc/samba/smb.conf
```

The **smb.conf** file is packed with detailed information on the Samba configuration (it's a good idea to read through it all). Scroll to the end of the file and add the configuration code from the **smb-pi.conf** listing. Press **CTRL+O** to save the file, and **CTRL+X** to exit nano.

File Explorer displaying the Samba share in its Network



05 Looking at the configuration file

The Samba configuration file uses the same format as Windows .ini files, with sections surrounded with square brackets. The commands are case-insensitive and ignore white space. Let's take a closer look at some of the lines in the file…

The location of the shared folder:

```
path = /home/pi/shared
```

Set this to 'no' to turn off the service:

```
available = yes
```

This is the list of users allowed to log in to the service:

```
valid users = pi
```

Whether this share is seen in the list of available shares in a net view and in the browse list:

```
browsable = yes
```

Ensures that users can write to files (the default is no):

```
writable = yes
```

Take a look at the Samba.org documentation for more information on all the configuration options

available in **smb.conf** (**magpi.cc/alEJZY**). Test out the Samba configuration file with testparm. Enter the following:

```
testparm
```

…and you should see a series of 'Processing section' tests. At the end it should say 'Loaded services file OK'. If not, head back to the start of this step and check you have entered the additional setup text correctly.

06 Create a password

Before we start the server, you'll want to set a Samba password. Enter:

```
sudo smbpasswd -a pi
```

Enter a simple and memorable password for testing (you can create a more secure one later by rerunning the command at any time).

Now restart your Samba server:

```
sudo service smbd restart
```

The Samba service will restart and the shared drive will now be available on the network.

07 Turn on SMB in Windows

Samba used to be installed by default in Windows, but it has recently become an optional installation.

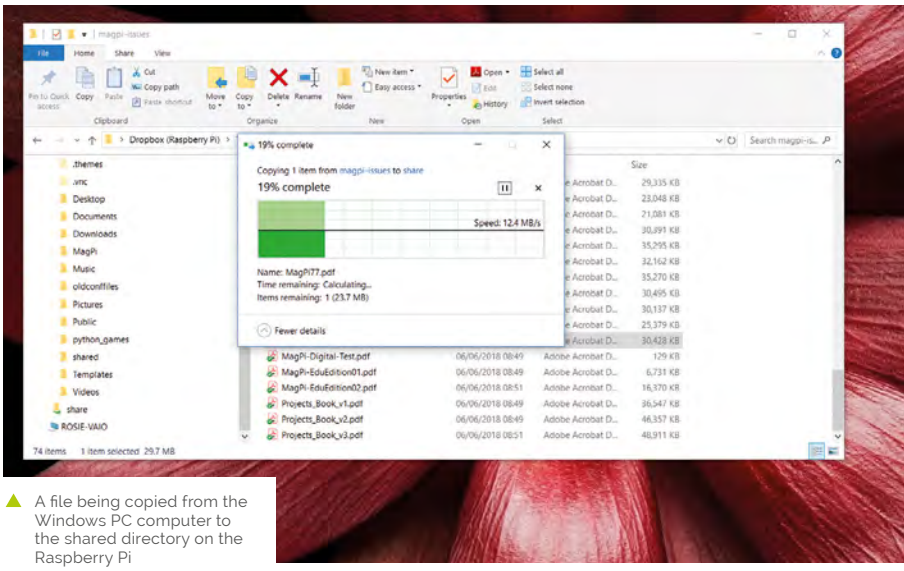
In Windows 10, click on the Search bar and type 'Control Panel'. Click on Control Panel in the search results. Now click 'Programs' and 'Turn Windows features on or off'. Scroll down to find 'SMB 1.0/CIFS File Sharing Support' and click the '+' expand icon to reveal its options. Place a check in the box marked 'SMB 1.0/CIFS Client'. Click OK. This will enable Samba client support on your Windows 10 PC so it can access the Raspberry Pi.

08 Find your Pi on the network

You'll now be able to find your Raspberry Pi file server (named RASPBERRYPI by default) from any device on your local network.

Click on the File Explorer icon in Windows and choose the Network option in the sidebar. You will see RASPBERRYPI in the list of devices. Click it and the 'Enter network credentials' window will appear. Enter your username **pi** and the password you created in Step 6. Click 'Remember my credentials' if you want to save the password, and then click OK.

The File Explorer will now display the shared directory. You can drag files and folders from Windows to the shared folder and they will be copied to your Raspberry Pi.



▲ A file being copied from the Windows PC computer to the shared directory on the Raspberry Pi

smb-pi.conf

DOWNLOAD THE FULL CODE: magpi.cc/pFBYLi

> Language: **ASCII**

```
001. [share]
002.     path = /home/pi/shared
003.     available = yes
004.     valid users = pi
005.     read only = no
006.     browsable = yes
007.     public = yes
```

09 Find the Pi on a Mac

Samba works across many different devices. And it's perfectly possible to access the shared folder from a Mac computer, or even an iOS or Android device (with the right app). Open a new Finder window on a Mac and choose Network in the sidebar. Click on 'raspberrypi' and 'shared' to access the directory. As with Windows, you can drag files directly to and from the Raspberry Pi.

Top Tip

Reboot everything

If you've set it all up and have checked everything, try restarting your Raspberry Pi and Windows computer. Sometimes this helps Windows find the drive on the network (once the shared drive appears, we find it stable).

Part 03

Code an isometric adventure game: AmazeBalls

Pygame Zero in 3D. Let's make some baddies and dynamite in this last part of the series

MAKER

Mark Vanstone

Educational software author from the nineties, author of the ArcVenture series, disappeared into the corporate software wasteland. Rescued by the Raspberry Pi!

magpi.cc/YiZnxL

@mindexplorers

We'll start from where we left off in the last part and add some extra elements to make a more challenging game.

We're going to add some baddie balls that roam around the maze, pushing walls about – so even if you know how to get to the finish, you may find your path is blocked. To give our player an antidote to being blocked in, we'll add some dynamite for them to pick up and use.

You'll Need

- Raspbian Jessie or newer
- Tiled (free map editor) **mapeditor.org**
- An image manipulation program such as GIMP, or images available from **magpi.cc/fPBrhM**
- The latest version of Pygame Zero (1.2)

01 Changing colours

Previously we had our ball bouncing around the maze by moving the drawing position of the map so that we are always viewing the area around the ball. Now we'll add some more balls, but these will be working against the player so we need to make them a different colour. We can do this quite easily with a paint app like GIMP. Just load each frame and use a tool called 'colorize' (in the Colors menu in GIMP). Make sure you save the frames as a different name; for example, put an 'e' for enemy in front of each file name.

02 Recycling code

We already have one ball bouncing around the maze – to get more balls, we'll try to reuse the code that we already have. We can duplicate the dictionary data at the top of our code that we have for the player and call it `enemy1` instead of `player`. You will want to change the `x` and `y` values in the data to something like 13, which will put the enemy ball near the middle of the maze. Now that we have our enemy defined, we can recycle some code.

03 Common update code

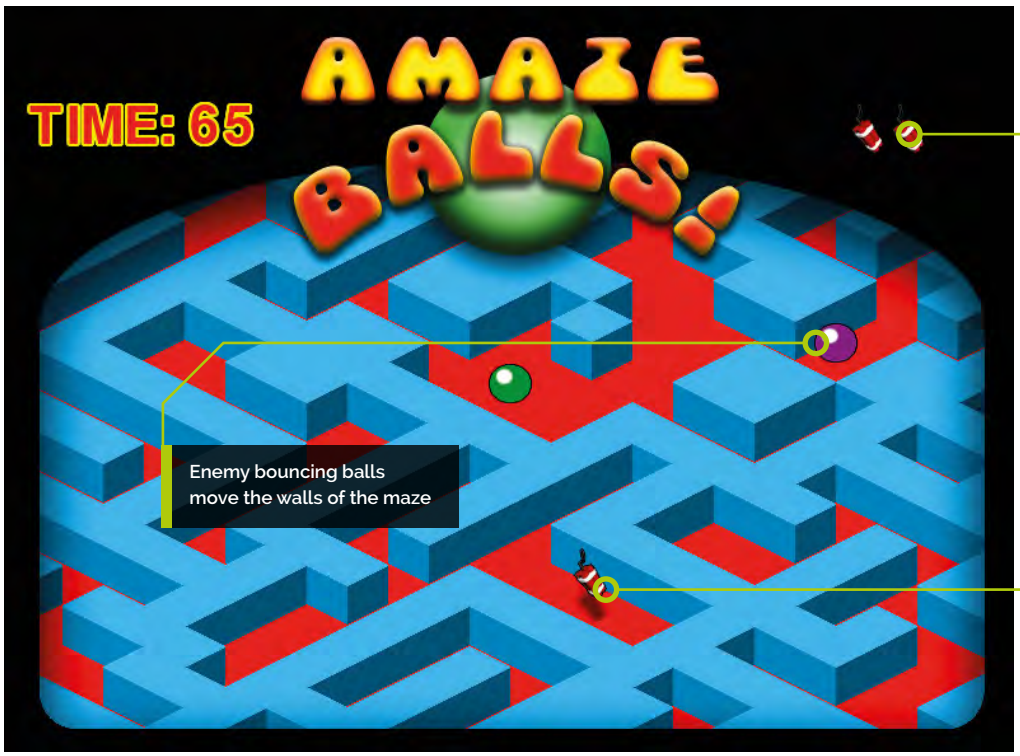
We can use exactly the same `updateBall()` function as we do for the player, and that will deal with all the animation and movement of the enemy from one block to the next. All we need to do is add another call to `updateBall()` after the one we have in our Pygame Zero `update()` function. But this time, rather than passing the player dictionary to the function, we pass the `enemy1` dictionary by writing `updateBall(enemy1)`. This means that if we set our enemy ball moving, all the changes to the data will be done in the same way as the player ball.

04 Drawing the enemy

Although we now have a way to update the animation of our enemy ball, we also need to write some code to draw it on the screen. This needs to be a bit different than the player ball because the enemy ball needs to move as the map is scrolled. We have to use both the map position and the `sx` and `sy` values of the ball to work out where it needs to be drawn. See **figure1.py** for the updates to the `drawMap()` function. You will see we are calculating the block position (`bx` and `by`) and then adding the `sx` and `sy` values.

05 Enemy brains

If we run our program now, we should see an enemy ball bouncing away in the middle of the maze. You'll notice that both balls bounce at exactly the same time – if you wanted to have the bounces non-synchronised, you could change the


TIME: 65
AMAZE BALLS!
Collected dynamite can be used to demolish walls that block the player
The player can pick up sticks of dynamite
Enemy bouncing balls move the walls of the maze

▼ Changes to `drawMap()` to incorporate the enemy ball

initial frame value where `enemy1` is declared at the top of the code. Now let's define a function called `updateEnemy()`; this will make the enemy ball move and also push some walls around.

06 Getting random

We are going to get the enemy to move around in a random way so, as we have done previously in this series, let's use the random module to generate some random movement. At the top of our program we import the module with `from random import randint`. Let's define our `updateEnemy()` function as `def updateEnemy(e):`. The `e` variable is the enemy dictionary that we will pass into the function when we call it. Now let's define some directions. We can do this with a list of x and y directions; for example, if we had x and y written as `[0,1]`, that would mean move no blocks in the x direction and one block in the y direction.

07 One direction

So, we can define all four directions as `edirs = [[-1,0],[0,1],[1,0],[0,-1]]`. And then all we need to do is pick one of them with a random number. To choose a random integer between 0 and 3, we write `r = randint(0,3)`. Now we can reuse

figure1.py

▶ Language: **Python 3**

```
001. def drawMap():
002.     psx = OFFSETX
003.     psy = OFFSETY-32
004.     mx = psx - player["sx"]
005.     my = psy - player["sy"]+32
006.
007.     for x in range(player["x"]-12, player["x"]+16):
008.         for y in range(player["y"]-12, player["y"]+16):
009.             if onMap(x,y):
010.                 b = mapData["data"][y][x]
011.                 td = findData(mapData["tiles"], "id", b)
012.                 block = td["image"]
013.                 bheight =  td["imageheight"]-34
014.                 bx = (x*32)-(y*32) + mx
015.                 by = (y*16)+(x*16) + my
016.                 if -32 <= bx < 800 and 100 <= by < 620:
017.                     screen.blit(block, (bx, by - bheight))
018.                 if x == player["x"] and y == player["y"]:
019.                     screen.blit("ball"+str(player["frame"]),
    (psx, psy))
020.                 if x == enemy1["x"] and y == enemy1["y"]:
021.                     screen.blit("eball"+str(enemy1[
    "frame"]),(bx + enemy1["sx"],(by-32)+enemy1["sy"]))
```

figure2.py

Language: **Python 3**

```
001.  def doMove(p, x, y):
002.      global mazeSolved
003.      if onMap(p["x"]+x, p["y"]+y):
004.          mt = mapData["data"][p["y"]+y][p["x"]+x]
005.          if mt == 1 or mt == 3:
006.              p.update({"queueX":x, "queueY":y,
    "moveDone":False})
007.              if mt == 3 and p == player:
008.                  mazeSolved = True
009.          return mt
```

▲ Changes to `doMove()` to make sure that the enemy ball doesn't trigger the finish condition

Top Tip

Dynamic map data

You can change any block on the map by changing the `id` in `mapData`. You could have lots of fun animating map elements in the `update()` function.

▲ There are several ways of creating images for games. This dynamite was created in a free 3D modelling program called Blender

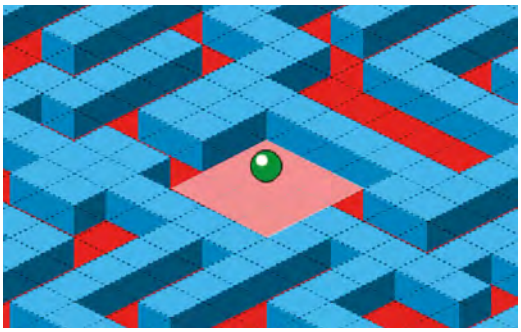
the `doMove()` function that moves the player, but use it for our enemy. We do need to make a couple of alterations to the `doMove()` function. The first is to return the `id` of the block that is being moved to. Then we can detect if a wall block is in the way; if it is, we can really mess things up for the player by moving the blocks around – sneaky, huh?

08 Making a move

The other problem that we have with the `doMove()` function is that it detects if the player has landed on the finish and if we use the same code for the enemy, we may get the finished condition triggered by the enemy instead of the player, so we need to change the `mazeSolved` condition to include a test to see if it's the player we are dealing with. Have a look at **figure2.py** to see these two changes to the `doMove()` function. When we've made those changes, we can go back to our `updateEnemy()` function.

09 Dynamic blocks

So far, all the blocks have stayed in the same place, but because they're all represented as numbers in our map data, we can change the numbers in the data and we'll see the maze change on screen. For example, if a block has `id` 2, it is a wall block. If we change the data to say that it is `id` 1, we'll see the wall disappear and there will be a floor block in its place. So, we can make changes to the maze as the enemy ball bounces around.



▲ When the dynamite is used, it changes all the blocks around the player (shown in pink here) to floor blocks

10 Moving the walls

So, going back to our `updateEnemy()` function, we need to first generate a random number for our direction and then move the enemy ball in that direction. But if it moves towards a wall, then we attempt to move that wall block in the direction the enemy ball is moving. The other thing we need to check is that there is a space for the wall to move into. We need to make a call to `doMove()` using the `enemy1` dictionary (passed into `updateEnemy()`) and then, if the block `id` is 2 (a wall), call another function called `moveBlock()`.

11 Changing places

We need to define the `moveBlock()` function and we will pass it the x and y block position in the map data and also the direction values that we are using to move the enemy ball. First, the function will check that we are moving data within the map area and then it will check that the block we are moving the wall to is a floor block (`id` 1). If this all checks out, then we copy the `id` of the block we are moving to the new position. Have a look at **figure3.py** to see the `updateEnemy()` function and the `moveBlock()` function.

12 What are we doing?

The **figure3.py** code may look a bit daunting in places, so let's have a look at the detail. The `updateEnemy()` function is basically defining four directions to move in and then we are saying: if the enemy is not currently moving, then get a random direction and pass the x and y values of that direction to the `doMove()` function. If the block we are moving to is `id` 2, then move the block using the location we are moving to and the direction values. Then zero the screen x and y coordinates (`sx` and `sy`) of the enemy dictionary.

13 It's all relative

You will notice that if the enemy is moving, then we check to see if we are on frame seven (when the ball actually moves from one block to another in the data) and if so, we fix up the coordinates so they are now relative to the new map location rather than the old one. The `moveBlock()` function just checks directly with the `mapData` data to check that the block can be moved, moves the data from the source location to the target location, and sets the source location to be a floor block.

14 Multiple enemies!

When all that is done, we just need to add `updateEnemy(enemy1)` after our `updateBall()` calls in the Pygame Zero `update()` function. Now, it may be that we consider that one enemy ball is not enough to make the game interesting and to make a second one is very easy now. We just need to duplicate the `enemy1` dictionary and call it `enemy2`, change the starting `x` and `y` to perhaps 25, make calls to `updateBall(enemy2)` and `updateEnemy(enemy2)` in the `update()` function and before you know it you have a second enemy ball. You could make as many as you like, or maybe put them in a list to be more efficient if there are more than three.

15 A bit one-sided

Now that we have our baddies messing up our maze, it's going to get pretty difficult for our player to get through to the finish, so it's time to level the playing field, in this case quite literally. Let's introduce some dynamite into the mix! We'll need to make a tile graphic for the dynamite that we can use in the Tiled map editor, and also an icon that we can use to show how many sticks of dynamite the player has collected. If you want to use ready-made graphics and map data, they are available from the GitHub repo: **magpi.cc/NvafjA**.

16 Handling the explosives

First, let's add a variable to hold the number of sticks of dynamite being held by the player, which can be done by writing `"dynamite":0` as part of the `player` dictionary. Then, assuming that we

figure3.py

Language: **Python 3**

```
001. def updateEnemy(e):
002.     edirs = [[-1,0],[0,1],[1,0],[0,-1]]
003.     if e["moveX"] == 0 and e["moveY"] == 0:
004.         r = randint(0,3)
005.         if doMove(e, edirs[r][0], edirs[r][1]) == 2:
006.             moveBlock(e["x"]+edirs[r][0],e["y"]+edirs[r]
     [1],edirs[r][0],edirs[r][1])
007.         e["sx"] = e["sy"] = 0
008.     else:
009.         if e["frame"] == 7 and e["movingNow"] == True:
010.             if e["sx"] == 12: e["sx"] -= 32
011.             if e["sx"] == -12: e["sx"] += 32
012.             if e["sy"] == 6: e["sy"] -= 16
013.             if e["sy"] == -6: e["sy"] += 16
014.
015. def moveBlock(mx,my,dx,dy):
016.     if onMap(mx+dx,my+dy):
017.         d = mapData["data"][my+dy][mx+dx]
018.         if d == 1:
019.             mapData["data"][my+dy][mx+dx] =
     mapData["data"][my][mx]
020.             mapData["data"][my][mx] = 1
```

▲ Updating the enemy ball and moving blocks if walls are in the way

▼ The updated `update()` function to include two enemy balls

figure4.py

Language: **Python 3**

```
001. def update(): # Pygame Zero update function
002.     global player, timer
003.     mt = 0
004.     if player["moveDone"] == True:
005.         if keyboard.left: mt = doMove(player, -1, 0)
006.         if keyboard.right: mt = doMove(player, 1, 0)
007.         if keyboard.up: mt = doMove(player, 0, -1)
008.         if keyboard.down: mt = doMove(player, 0, 1)
009.     if mt == 4:
010.         mapData["data"][ player["y"] + player["queueY"]][
     player["x"] + player["queueX"]] = 1
011.         player["dynamite"] += 1
012.     updateBall(player)
013.     updateBall(enemy1)
014.     updateBall(enemy2)
015.     updateEnemy(enemy1)
016.     updateEnemy(enemy2)
```

have added some dynamite to our map data (see the previous part of this series for details on editing with Tiled), we need to detect if our player has moved onto a dynamite block and, if so, add `1` to our `dynamite` count and make the dynamite block into a floor block, which will make it disappear from the map. We can do this in our `update()` function.

17 Stockpiling ammo

To handle the picking up, we just need to test the value of `mt` after our keyboard checks. See **figure4.py** to view the revised `update()` function. When our player has picked up some dynamite, we can display the number held with icons, as we have done before (for example with lives), in the `draw()` function by writing `for l in range(player["dynamite"]): screen.blit("dmicon", (650+(l*32),80))`, which will draw our dynamite icons in the top right of the screen. So now we have the ammunition for our player to blow a path through the blockages that the baddies have put in the way.

18 Going off with a bang

All we have left to do now is to code a mechanism to set off the dynamite. We will do this with the Pygame Zero `on_key_down()` function. We need to test if the **SPACE** bar has been pressed and, if so, clear a space around the player, making all the blocks into floor blocks. This can be done with a nested `for` loop. Have a look at the full **amazeballs3.py** listing to see this last bit of code.

Now is the time to test how the game is set up, is it too easy or too hard? Do you need more or fewer enemies? You could try making a range of different maps with other objects to collect.

19 And finally

Well, sadly that's all we have time for in this series. We hope you have learned a lot about Pygame Zero and writing games in Python. We must at this stage give a big shout out to the creator of Pygame Zero, Daniel Pope: without his excellent work, this series would not have existed. We hope you would agree that the Pygame Zero framework is an ideal starting place to learn game coding on the Raspberry Pi.

amazeballs3.py

▶ Language: **Python**

```
001. import pgzrun
002. import map3d
003. from random import randint
004.
005. player = {"x":3, "y":3, "frame":0, "sx":0, "sy":96,
006.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
007.           "moveDone":True, "movingNow":False,
     "animCounter":0, "dynamite":0}
008. enemy1 = {"x":13, "y":13, "frame":0, "sx":0, "sy":0,
009.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
010.           "moveDone":True, "movingNow":False,
     "animCounter":0}
011. enemy2 = {"x":25, "y":25, "frame":0, "sx":0, "sy":0,
012.           "moveX":0, "moveY":0, "queueX":0, "queueY":0,
013.           "moveDone":True, "movingNow":False,
     "animCounter":0}
014. OFFSETX = 368
015. OFFSETY = 300
016. timer = 0
017. mazeSolved = False
018.
019. mapData = map3d.loadmap("maps/map1.json")
020.
021. def draw(): # Pygame Zero draw function
022.     screen.fill((0, 0, 0))
023.     drawMap()
024.     screen.blit('title', (0, 0))
025.     screen.draw.text("TIME: "+str(timer) , topleft=(20,
     80), owidth=0.5, ocolor=(255,255,0), color=(255,0,0) ,
     fontsize=60)
026.     for l in range(player["dynamite"]): screen.blit(
     "dmicon", (650+(l*32),80))
027.     if mazeSolved:
028.         screen.draw.text("MAZE SOLVED in " + str(timer) + "
     seconds!" , center=(400, 450), owidth=0.5, ocolor=(0,0,0),
     color=(0,255,0) , fontsize=60)
029.
030.
031. def update(): # Pygame Zero update function
032.     global player, timer
033.     mt = 0
034.     if player["moveDone"] == True:
035.         if keyboard.left: mt = doMove(player, -1, 0)
036.         if keyboard.right: mt = doMove(player, 1, 0)
037.         if keyboard.up: mt = doMove(player, 0, -1)
038.         if keyboard.down: mt = doMove(player, 0, 1)
039.     if mt == 4:
040.         mapData["data"][ player["y"] + player["queueY"]][
     player["x"] + player["queueX"]] = 1
041.         player["dynamite"] += 1
042.     updateBall(player)
043.     updateBall(enemy1)
044.     updateBall(enemy2)
045.     updateEnemy(enemy1)
046.     updateEnemy(enemy2)
047.
048. def on_key_down(key):
049.     if player["dynamite"] > 0 and key.name == "SPACE":
050.         player["dynamite"] -= 1
051.         for x in range(player["x"]-1, player["x"]+2):
```

DOWNLOAD THE FULL CODE:
magpi.cc/ODgUCg

```
            for y in range(player["y"]-1, player["y"]+2):
                mapData["data"][y][x] = 1

def timerTick():
    global timer
    if not mazeSolved:
        timer += 1

def drawMap():
    psx = OFFSETX
    psy = OFFSETY-32
    mx = psx - player["sx"]
    my = psy - player["sy"]+32

    for x in range(player["x"]-12, player["x"]+16):
        for y in range(player["y"]-12, player["y"]+16):
            if onMap(x,y):
                b = mapData["data"][y][x]
                td = findData(mapData["tiles"], "id", b)
                block = td["image"]
                bheight =  td["imageheight"]-34
                bx = (x*32)-(y*32) + mx
                by = (y*16)+(x*16) + my
                if -32 <= bx < 800 and 100 <= by < 620:
                    screen.blit(block, (bx, by -
bheight))
                if x == player["x"] and y == player["y"]:
                    screen.blit("ball"+str(player[
"frame"]), (psx, psy))
                if x == enemy1["x"] and y ==
enemy1["y"]:
                    screen.blit("eball"+str(enemy1[
"frame"]), (bx + enemy1["sx"], (by-32)+enemy1["sy"]))
                if x == enemy2["x"] and y ==
enemy2["y"]:
                    screen.blit("eball"+str(enemy2[
"frame"]), (bx + enemy2["sx"], (by-32)+enemy2["sy"]))

def findData(lst, key, value):
    for i, dic in enumerate(lst):
        if dic[key] == value:
            return dic
    return -1

def onMap(x,y):
    if 0 <= x < mapData["width"] and 0 <= y <
mapData["height"]:
        return True
    return False

def doMove(p, x, y):
    global mazeSolved
    if onMap(p["x"]+x, p["y"]+y):
        mt = mapData["data"][p["y"]+y][p["x"]+x]
        if mt == 1 or mt == 3 or mt == 4:
            p.update({"queueX":x, "queueY":y,
"moveDone":False})
            if mt == 3 and p == player:
                mazeSolved = True
        return mt


def updateEnemy(e):
    edirs = [[-1,0],[0,1],[1,0],[0,-1]]
    if e["moveX"] == 0 and e["moveY"] == 0:
        r = randint(0,3)
        if doMove(e, edirs[r][0], edirs[r][1]) == 2:
            moveBlock(e["x"]+edirs[r][0],e["y"]+
edirs[r][1],edirs[r][0],edirs[r][1])
        e["sx"] = e["sy"] = 0
    else:
        if e["frame"] == 7 and e["movingNow"] == True:
            if e["sx"] == 12: e["sx"] -= 32
            if e["sx"] == -12: e["sx"] += 32
            if e["sy"] == 6: e["sy"] -= 16
            if e["sy"] == -6: e["sy"] += 16

def moveBlock(mx,my,dx,dy):
    if onMap(mx+dx,my+dy):
        d = mapData["data"][my+dy][mx+dx]
        if d == 1:
            mapData["data"][my+dy][mx+dx] =
mapData["data"][my][mx]
            mapData["data"][my][mx] = 1

def updateBall(p):
    if p["movingNow"]:
        if p["moveX"] == -1: moveP(p,-1,-0.5)
        if p["moveX"] == 1: moveP(p,1,0.5)
        if p["moveY"] == -1: moveP(p,1,-0.5)
        if p["moveY"] == 1: moveP(p,-1,0.5)
    p["animCounter"] += 1
    if p["animCounter"] == 4:
        p["animCounter"] = 0
        p["frame"] += 1
        if p["frame"] > 7:
            p["frame"] = 0
        if p["frame"] == 4:
            if p["moveDone"] == False:
                if p["queueX"] != 0 or p["queueY"] !=0:
                    p.update({"moveX":p["queueX"],
"moveY":p["queueY"], "queueX":0, "queueY":0,
"movingNow": True})
                else:
                    p.update({"moveDone":True, "moveX":0,
"moveY":0, "movingNow":False})

        if p["frame"] == 7 and p["moveDone"] == False
and p["movingNow"] == True:
            p["x"] += p["moveX"]
            p["y"] += p["moveY"]
            p["moveDone"] = True

def moveP(p,x,y):
    p["sx"] += x
    p["sy"] += y

clock.schedule_interval(timerTick, 1.0)
pgzrun.go()
```

Make comics from TV recordings

Convert a recording from the Raspberry Pi TV HAT into a comic book and read the latest Doctor Who episode on your Kindle

MAKER

PJ Evans

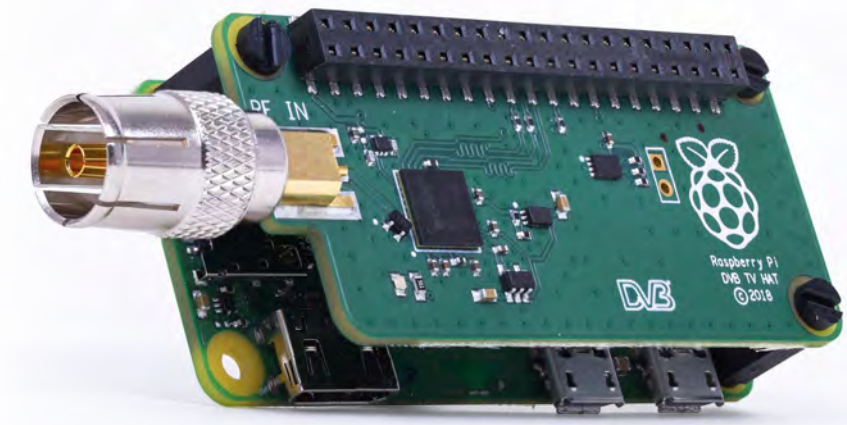
PJ is a writer, developer, and Milton Keynes Jam wrangler. He has terrible taste in movies.

mrpjevans.com

Read any good telly recently? How about catching up on your favourites shows on an e-ink reader or tablet? Sounds silly but it can be a nice, peaceful alternative to sit back and flick through *Holby City*. OK, we're reaching a little bit. The real fun here is learning about video and image manipulation, optical character recognition, generating PDFs in code, and using Python as a powerful scripting language to pull several tools together. We'll take the raw recording produced by the Raspberry Pi TV HAT and create a PDF document, complete with captions taken from subtitles.

01 Get recording

Before starting, make sure you have your Raspberry Pi set up with a TV HAT and Tvheadend installed (see 'You'll Need' box for a helpful link). You will need a recording from Tvheadend (it doesn't matter what, but maybe the news wouldn't be the most exciting choice). You can select any programme and record it, then find the recording under 'Digital Video Recorder' then 'Finished Recordings'. From here you can download the file or you can find recordings in **/var/lib/hts**. Tvheadend records in the original broadcast MPEG-2 TS format (or 'transport stream').



▼ The TV HAT connects to a digital antenna, giving your Raspberry Pi over-the-air access to over 80 TV channels and radio

02 Install dependencies

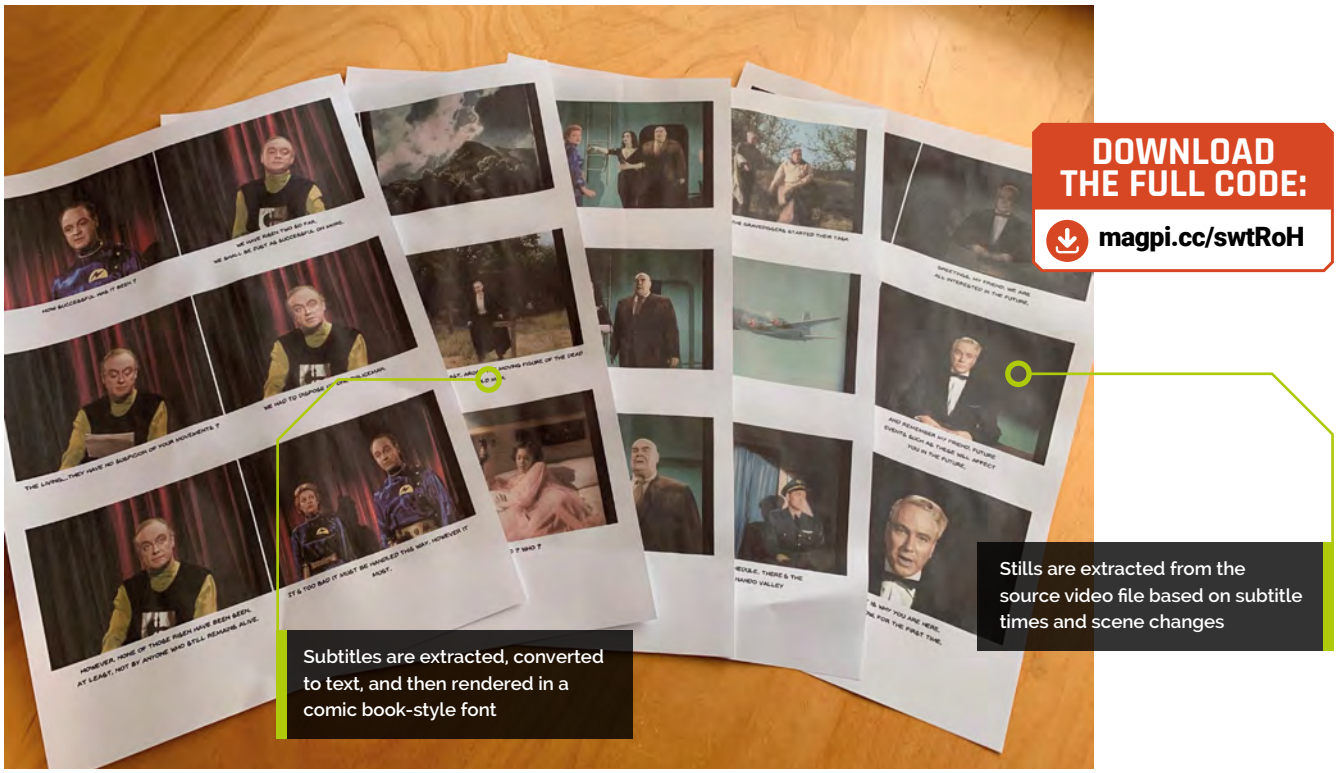
The process of converting a recording to a PDF is going to take several discrete stages. These include video extraction, optical character recognition (OCR), and generating PDFs. Not all of this is easily within Python's reach, so we'll use Python to manage the process, delegating the 'heavy lifting' to some command-line utilities. Their purposes will become apparent as we go through the tutorial. Here's what you need to do at the command-line:

```
  sudo apt update && sudo apt -y upgrade
  sudo apt install git python3-pip ffmpeg
imagemagick
  pip3 install fpdf arrow
```

03 Compile and install ccextractor

The utility 'ccextractor' is able to remove subtitles from DVB (Digital Video Broadcasting) recordings. Unfortunately, this application is not available in the APT repositories, so we're going to have to compile it ourselves. We'll use Git, which we installed in the previous step, to download the source code from its repository. Then we'll install its dependencies (other programs it relies on) before compiling and installing the app.

```
  cd
  git clone https://github.com/CCExtractor/
```


DOWNLOAD THE FULL CODE:
magpi.cc/swtRoH
Subtitles are extracted, converted to text, and then rendered in a comic book-style font
Stills are extracted from the source video file based on subtitle times and scene changes

```
ccextractor.git
  sudo apt install -y libglfw3-dev cmake
gcc libcurl4-gnutls-dev tesseract-ocr
tesseract-ocr-dev libleptonica-dev
  cd ccextractor/linux
  ./build
  sudo mv ./ccextractor /usr/local/bin/
```

04 Install the script

As this is a series of steps potentially involving hundreds if not thousands of files, we've provided a Python script to control the process. It's a bit on the large side to type in manually, so again we'll use Git. To get the code on to your Pi, enter the following commands:

```
  cd
  git clone https://github.com/mrpjevans/
comical.git
```

You will now have a new directory, **comical**, containing the script and a few other files we need.

05 Extract the subtitles

Rather than just run the entire script, which wouldn't show us much, let's run it in stages. Make sure you know the path to your recording. We're using the public domain movie *Plan 9 From Outer Space*, regarded as one of the worst films ever made. The first job is to extract the subtitles from the video so we can process them and use them as captions.

> Before starting, make sure you have your Raspberry Pi set up with a TV HAT and Tvheadend installed

```
cd ~/comical
python3 comical.py -i plan9.ts --extract
```

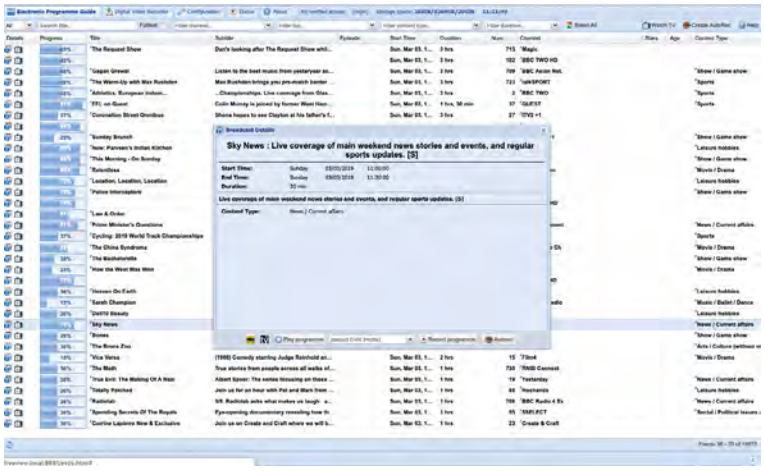
A folder, **plan9.d**, is created, containing a PNG image file for each subtitle. An XML file, **plan9.xml**, contains the timing information for each title.

06 Cleaning up

So why are our subtitles images? It's because that's the European digital broadcast standard. Subtitles in DVB are actually a second video stream. To make use of them, we'll need to take the PNGs that ccextractor created and perform

You'll Need

- Raspberry Pi TV HAT **magpi.cc/oBXuot**
- Tvheadend installation **magpi.cc/QCkFdt**
- e-book reader or tablet



▲ Strange name and strange interface, but Tvheadend more than makes up for that with its seemingly endless features

optical character recognition on them. Currently they're too small to be recognised accurately by the OCR application Tesseract. So, we'll use the ImageMagick utility 'mogrify' to resize them and greyscale them.

```
python3 comical.py -i plan9.ts --clean
```

If you have a look in the directory, you'll see the subtitles are now large and monochrome.

07 OCR With Tesseract

Tesseract is a remarkable utility originally developed by Hewlett-Packard and open-sourced. Now you have it installed on your Pi, you can use it for many other purposes. To convert something into text, just run:

```
tesseract <image file> <output file>
```

Our script reads in every image in the directory and sends it to Tesseract for processing. At this size, you can expect a good level of accuracy from DVB titles.

```
python3 comical.py -i plan9.ts --ocr
```

In the same directory you'll now see a matching '.txt' file for each graphic subtitle.

08 Extract images

The next part of our script will extract a single still image for each subtitle based on that subtitle's timestamp. To get the timestamps, we use Python's built-in XML parsing libraries. For each timestamp, we then ask ffmpeg (a Swiss Army knife for video processing) to extract a still image as a JPEG and save it in (in our case) a new directory called **plan9_process**. The file name represents the time code at which it appears. We also copy across the subtitle with a matching file name.

```
python3 comical.py -i plan9.ts --images
```

09 A picture worth a thousand words

Have a look in your equivalent of **plan9_process**. We've got everything we need to build our PDF. Right? Well, yes, provided there's no break in dialogue, which seems unlikely. What about scenes with no subtitles? Again, ffmpeg comes to our rescue. An advanced filter can detect when a significant amount of the screen changes, denoting a scene change. Our script will ask ffmpeg to detect every scene change and then extract further JPEG images, ignoring any that are within a second of a subtitle image.

```
 python3 comical.py -i plan9.ts
--detectscenes
 python3 comical.py -i plan9.ts
--extractscenes
```

Your **_process** directory is now populated with the additional images.

10 Build it!

The final part of the script will take all the images and text files and convert them into a PDF for you to enjoy.

```
 python3 comical.py -i plan9.ts -o plan9.
pdf --build
```

This part of the script uses the fpdf Python library to lay out each image in a 2×3 grid, adding pages as needed. Where there is a matching subtitle, it is placed below the image. To give the final result a bit more of a graphic novel feel, there is a comic book-style font included in the comical repository which is used by fpdf when rendering text.

Top Tip

Not just DVB

The script will also work with other video formats that have a supported video subtitle track.

> Tesseract is a remarkable utility originally developed by Hewlett-Packard and open-sourced

11 Adjusting fonts

You might find that sometimes, dependent on the subtitle lengths, the captions can overflow; or that the font size isn't large enough, with too much white space. The script provides a few arguments that can be specified on the command line to help with this:

```
python3 comical.py -i plan9.ts -o plan9.
pdf --build --fontsize 8 --lineheight 5
--offset 68
```

Here, `fontsize` sets the size of the font. This needs to be in step with `lineheight`, which sets the vertical spacing between lines. `offset` sets the position of the first line of text below the image. The default settings are shown above.

12 Automating and fine-tuning

The **comical.py** script comes with a number of arguments to control its behaviour. In the tutorial we've gone step by step, but you could have just run the following:

```
python3 comical.py -i plan9.ts -o plan9.
pdf --full
```

This performs every step in one operation. You can also do a pre-build:

```
python3 comical.py -i plan9.ts -o plan9.
pdf --prebuild
```

This performs every step except building the PDF, as you might want to remove unwanted images and subtitles to crop the PDF to the things in which you are interested. Delete the unwanted files from your **_process** directory and then run:

```
python3 comical.py -i plan9.ts -o plan9.
pdf --build
```

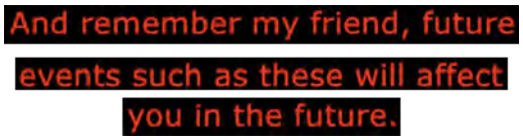
13 Make it your own

You could regard this project as a bit frivolous, but in the process of putting it together we've looked at several cool technologies such as video manipulation and optical character recognition. Examine the script code to see how we use Python to link all these different utilities together and marshal the data flowing between them. Why not see if you can improve on the results? Some ideas include adding filters to the images to give a graphic novel appearance, watching the recordings folder to trigger automatic conversion, creating glitch art, or mashing up different recordings.

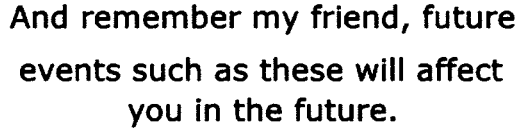
Top Tip

Get arty

Try using OpenCV to give each image a comic book look: **magpi.cc/RNVeoD**



This is an original subtitle. Yes, the script is really that bad. It's a PNG taken from the subtitle video stream



We use ImageMagick's mogrify utility to remove colour, invert the image, and increase size by 400% to improve OCR accuracy.

```
And remember my friend, future

events such as these will affect
you in the future.
```

Tesseract reads in the image and produces text output. As you can see, it's very accurate if the source material is clear

The final step is to produce a more fun caption for the panel by rendering it in a comic book font

Build a karaoke machine

With some clever software and a bit of equipment, you can turn a Raspberry Pi into a fully featured karaoke system

MAKER

K.G. Orphanides

K.G. is a writer, developer and sometime musician who specialises in getting strange noises out of Pi boards. A significant amount of Danzig was sung in the course of this build.

@KGOrphanides

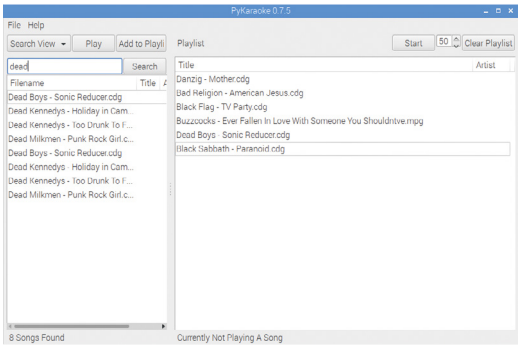
You'll Need

- HiFiBerry DAC+ ADC **magpi.cc/dQKrSL**
- Dynamic microphone (e.g. Shure SM-58 or QTX 173.853)
- XLR or TRS to 3.5 mm adapter
- Speakers with stereo RCA connectors
- Monitor, TV, or projector via HDMI
- MP3 + CGD karaoke tracks

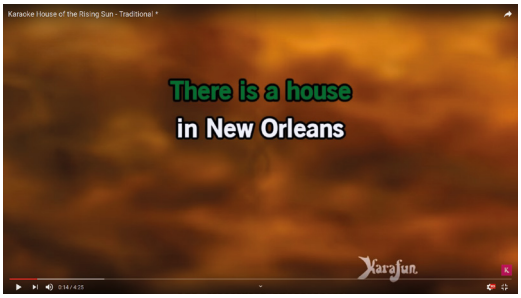
Karaoke can be a party entertainment highlight, a brilliant wet-weather kids' activity, or a workout for your vocal skills. There are two main karaoke packages that work well on the Pi. We're going to focus on the more accessible of these, PyKaraoke, as it's light on resources and easy to get up and running. However, if you'd like a more polished interface, see 'Get Professional with OpenKJ' box (overleaf).

Because the Pi doesn't have a mic input, we're using a HiFiBerry DAC+ ADC HAT, which provides a 3.5 mm stereo input with an optional pre-amp circuit and dual RCA outputs that you can connect to a pair of powered speakers. To go with that, you'll need a dynamic microphone, plus relevant cables to connect your mic's XLR or ¼-inch TRS plug down to a 3.5 mm male plug, a decent pair of speakers, and a big monitor, TV, or projector to display lyrics on.

We recommend starting this project with a fresh Raspbian install, fully updated and configured to expand its file system and match your display's overscan requirements using `sudo raspi-config`.



▲ PyKaraoke provides a simple, searchable interface to find and play your karaoke collection



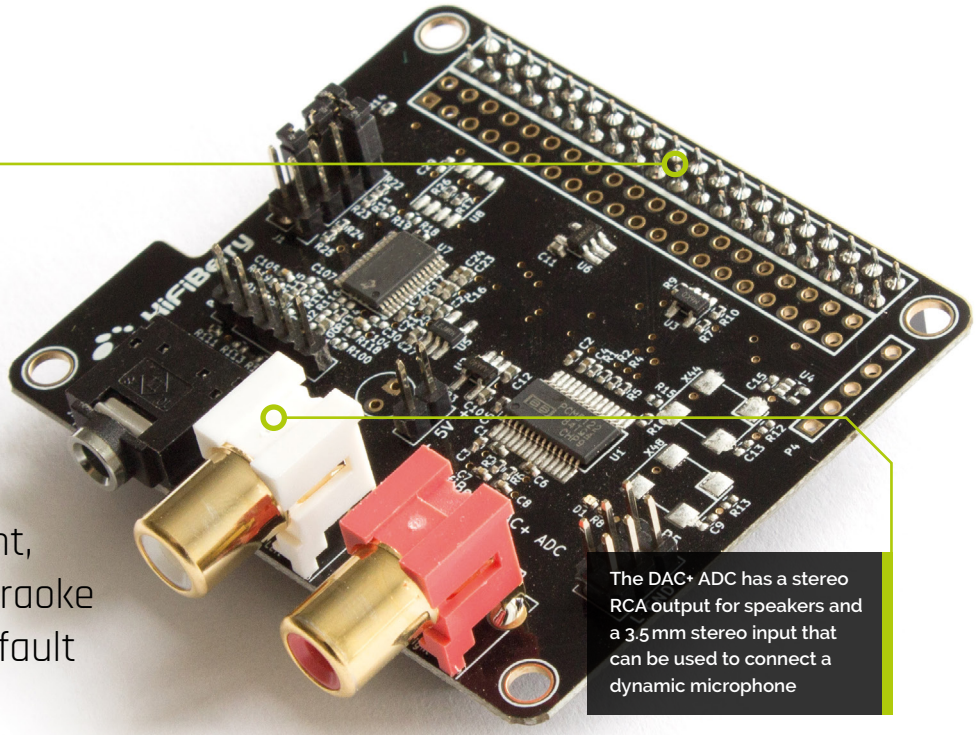
▲ If you don't want to collect your own tunes, you can always search for karaoke tracks on YouTube and sing along using a PulseAudio mic loopback

01 Connect the DAC+ ADC HAT

The HiFiBerry DAC+ ADC mounts on the Pi's GPIO. The daughterboard comes with a set of four optional plastic mounting spacers to keep both boards securely in place, although you'll want to leave these out if you plan on putting it into one of HiFiBerry's Universal Cases, due to space limitations. A specific DAC+ ADC case is due out in the coming months.

02 Configure mic input gain

The DAC+ ADC's 3.5 mm stereo input is by default configured to accept line-level audio – sound that's already been amplified, such you'd get from the line out connectors on most audio gear. To use a dynamic microphone with it, you'll have to set the input gain using a jumper switch on the J1 header. The jumper should cover the second pair of pins from the top of J1 to enable the DAC+ ADC's optional 32 dB gain.



The HAT sits on top of your GPIO, using GPIOs 2–3 for configuration and GPIOs 18–21 as an audio interface

The DAC+ ADC has a stereo RCA output for speakers and a 3.5mm stereo input that can be used to connect a dynamic microphone

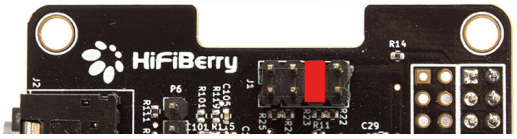
> PyKaraoke is an efficient, reliable, and no-frills karaoke player in Raspbian's default software repository

03 Update your kernel

Current versions of Raspbian Stretch, as of early 2019, don't ship with kernel support for some of the DAC+ ADC's features. Fortunately, HiFiBerry has produced a customised kernel version 4.19.y to replace Raspbian's kernel 4.14 and correctly update **/boot/config.txt** to load appropriate drivers for the audio device. In a Terminal window, enter `bash <(curl https://raw.githubusercontent.com/hifiberry/dacadckernel/master/update-kernel)` to begin the process, which can take a while. If you're unsure of your current kernel version, check by typing `uname -a`.

04 Connect and test your hardware

Plug your microphone, using whatever adapters it needs, into the 3.5mm port on the DAC+ ADC and connect your speakers to its RCA ports. HiFiBerry's upgrade tool disables the Pi's on-board Broadcom BCM2837 audio processor by default to help avoid conflicts, so only the DAC+ ADC should appear during these tests. Type `aplay -l` at the command line to list output devices, and `arecord -l` to view inputs. Type `speaker-test -c2` to run a white noise test of your left and right speaker channels, and type `alsaloop` to confirm that your mic input can be picked up and fed to the speakers.



The HiFiBerry DAC+ ADC has a built-in microphone pre-amp, which you can set the gain on via a jumper switch (32dB shown here)

05 Install PyKaraoke

PyKaraoke is an efficient, reliable, and no-frills karaoke player and, best of all, it's in Raspbian's default software repository. Just `sudo apt install pykaraoke` to install the program and all its dependencies. A shortcut for the PyKaraoke GUI will be placed in your Sound & Video menu. For font support on MIDI karaoke files and in the ultra-lightweight PyKaraoke Mini player, `sudo apt install ttf-dejavu`.

06 Set up your microphone loopback

Like many karaoke players, PyKaraoke doesn't have its own mic input handling, but

Top Tip

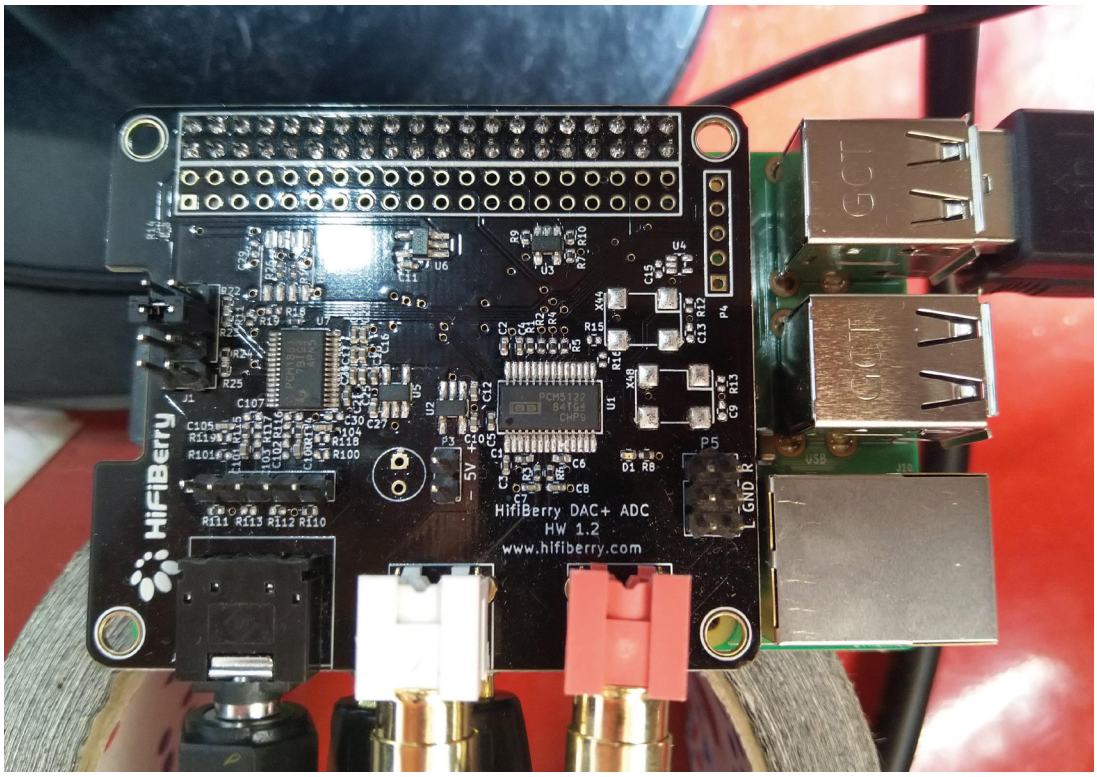
Sound control

For a handy graphic equaliser application to control your audio levels, sudo apt install qasmixer.

Top Tip

Dual audio

If you want to use the Pi's on-board audio via an HDMI link to a television or AV receiver and simultaneously use the audio HAT with a second pair of speakers for the mic, uncomment `dtparam=audio=on` in **/boot/config.txt**.

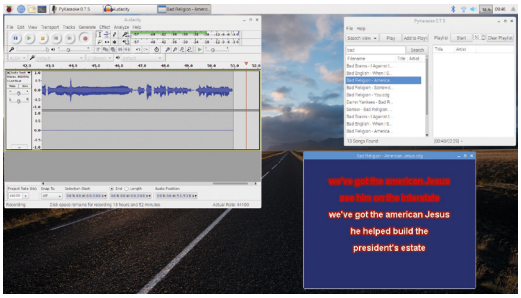


A Pi 3 with DAC+ ADC HAT, connected to powered speakers, an XLR mic via a 3.5mm connector and XLR to TRS cable, HDMI monitor, mouse, and keyboard

because it relies on the PulseAudio subsystem, we'll hit a device conflict if we try to use the ALSA loopback from our test. To avoid this, we can create a loopback using PulseAudio instead. Open a Terminal and `sudo apt install pulseaudio`. Now type `pactl load-module module-loopback latency_msec=1`. Check to make sure all's working as it should. Only run this command once per session, or multiple mic loopbacks will be created.

07 Find some sweet tunes

Before you can hold your grand karaoke night, you'll need something to sing. You can buy backing tracks online, but unless you're happy reading from a lyric sheet, make sure they come with a matching CDG on-screen lyric file. Commercial sites such as **karaoke-version.com** have limited free selections, and a number of musicians create karaoke backing tracks as donorware, putting them online via file-sharing services, social media groups, or by uploading them to YouTube for people to sing along with online or download using tools such as YouTube-downloader. You can even find some on the Internet Archive.



If you want to record your friends' performances for posterity, load up Audacity to capture the mic input as they sing

Where to find free karaoke backing tracks

Specialist sites:
magpi.cc/zikqUb
magpi.cc/zWYkfW

YouTube:
magpi.cc/uHMQaC
magpi.cc/frkEhY

Internet Archive:
magpi.cc/mtztmt

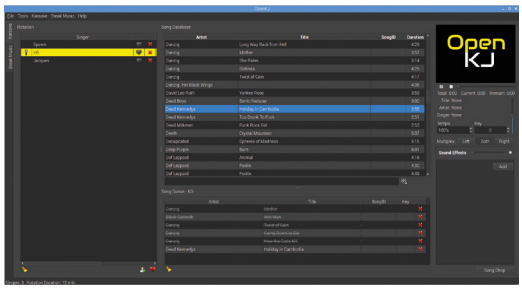
Facebook:
facebook.com/bostonmetalkaraoke

08 Copy in your karaoke files

The best-performing and most consistent karaoke format to work with uses specially created MP3 backing files with a matching CDG graphical lyrics file. PyKaraoke also supports formats including MPEG videos, MIDI files with integrated lyrics, and MIDI-based KAR files. Create a **Karaoke** directory in your home folder to contain your files – you might want to further separate these by format, artist or alphabetically.

09 Configure PyKaraoke

Open PyKaraoke go to File > Add new songs to database. Add the **Karaoke** folder you created in



▲ OpenKJ provides a much more sophisticated interface, but its high system load may result in performance issues in some files if you don't overclock and cool your Pi

the last step, click Scan Now and, when it's done, click Save and Close. There's also a Configure option in the File menu to tweak your other settings. Here, you can set the lyrics window to open at full-screen, set default sample rates, and choose an external video player to handle MPEG and AVI karaoke videos. While you may wish to change the default settings later, you can safely leave them alone to start with.

10 Ready, set, sing!

With all your files added, you can just type a word or letter into PyKaraoke's search bar to see everything that matches. Drag tracks to the playlist frame on the right to queue them up before hitting Start, or just click Play to play them immediately. Your mic's already live and ready to go, thanks to our PulseAudio loopback. If you're having trouble finding a song, click on the Search View button and switch to Folder View, where you can browse your entire file system for compatible files.

11 Loopback on load

Finally, to save a bit of effort, let's create a Bash script to automatically set up our mic loopback and open PyKaraoke. Use a text editor to create a new file called **karaoke** in your home directory. Our brief code box shows you how to format the commands. At the command line, `chmod u+x karaoke` to make it executable, and use the Main Menu editor to add it to Sound & Video as a new item for convenience.

Get professional with OpenKJ

While PyKaraoke is more than sufficient for singing along at home, if you want to run a more polished karaoke night, OpenKJ has a better interface, playlist queues for multiple singers, and a wide range of extra features.

It works, but sometimes pushes the limits of the Raspberry Pi 3 B+'s processor. If you'd like to give its enhanced features a go, follow these instructions to compile and install it from source.

Install audio dependencies:

```
sudo apt install pulseaudio
gstreamer1.0-pulseaudio
gstreamer1.0-tools gstreamer1.0-
plugins-good gstreamer1.0-
plugins-ugly gstreamer1.0-
plugins-bad libgstreamer-
plugins-base1.0-dev
```

Install QT5:

```
sudo apt install qt5-default
qtcreator qtmultimedia5-
dev libqt5multimediawidgets5
libqt5multimedia5-plugins
libqt5multimedia5 libqt5svg5-dev
qt5-qmake
```

Install libtag:

```
sudo apt install libtag1-dev
libtag-extras1 libtag-extras-dev

git clone https://github.com/
OpenKJ/OpenKJ.git/.
cd OpenKJ/OpenKJ
/usr/lib/arm-linux-gnueabihf/
qt5/bin/qmake
make

sudo apt install checkinstall
sudo checkinstall
```

Checkinstall will by default install OpenKJ and create a .deb package that you can use in future.

> The best karaoke format to work with uses specially created MP3 backing files with a matching CDG graphical lyrics file

karaoke

Language: **Bash**

```
001. #!/bin/bash
002.
003. pactl load-module module-loopback latency_msec=1
004. pykaraoke
```

▲ This brief script enables your mic loopback before launching the karaoke player

Pictures from space via ham radio

Have you ever wanted to receive a radio signal from space? It's fun and a lot easier than you might think!

MAKER

Dave Honess

Dave lives in the Netherlands and works at the European Space Research and Technology Centre, where he helps coordinate the educational operations for European crew on the ISS.

@dave_spice

What is Slow Scan TV (SSTV), then? You convert a picture into sound (similar to fax machine or dial-up modem noise) transmit the sound via radio, and convert the received sound back into a picture at the other end. The International Space Station (ISS) has a long history of transmitting SSTV pictures for global events, each lasting several days, where people the world over tune their radios to the ISS and try to receive the pictures every time it flies over their home. It's a hobby of mine!

One such event is set to begin on **12 April 2019** to celebrate the anniversary – to the day – of the launches of Yuri Gagarin in 1961 and the first Space Shuttle, 20 years later, in 1981.

You'll Need

- Raspberry Pi 2B or later
- 3.5 mm speakers or headphones
- RTL-SDR USB dongle (search for 'RTL2832U') with a good whip or dipole antenna (or just get this starter kit: **magpi.cc/uBMKQg**)
- Tape measure, at least 1 metre

01 Install prerequisites and set the time

We presume you are running a clean install of Raspbian Stretch Desktop edition and are online.

▲ This is one of the clearest pictures your author has been able to receive with this setup. It was from an 83 degree elevation pass near Amsterdam, in the Netherlands, on 15 February 2019

Firstly, switch the audio output to the 3.5 mm headphone jack. This is needed for the way PulseAudio loops back the system audio as an input device. On the Raspbian desktop, simply right-click the speaker icon in the menu bar and choose Analog.

Then, open a Terminal and install the required software for the project:

```
sudo apt-get update
sudo apt-get install rtl-sdr sox
pulseaudio qsstv ntpdate -y
```

Next, set your time zone and pull the correct time from the internet. In the Raspbian main menu, go to Preferences > Raspberry Pi Configuration > Localisation tab > Set Timezone, change Area and Location accordingly, then click OK.

Finally, go back to the Terminal window to pull the time:

```
sudo ntpdate pool.ntp.org
```

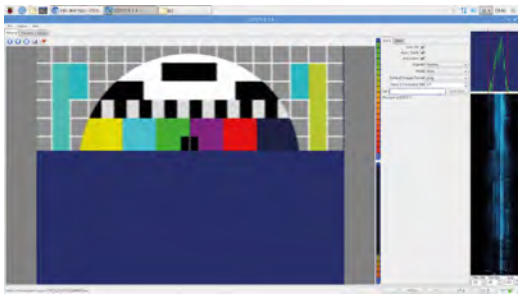
Note that the desktop clock in the top right only updates every minute.

02 Install and test the RTL-SDR dongle

Insert the RTL-SDR dongle into your Pi and screw in the antenna (it can sit on your desk at this stage). Then verify the RTL-SDR is working with the rtl_test program. In a Terminal window, enter:

```
rtl_test
```

If you see a 'failed to open rtlsdr device' error, remove the dongle, reinsert it, and try `rtl_test`



It takes about two whole minutes for a picture to be decoded. When they do this from the ISS, a picture is transmitted continuously for about two minutes, with two minutes of off-time before the next picture starts

again. Leave for about 30 seconds and look out for any messages about loss of samples. Press **CTRL+C** when 30 seconds has elapsed. The loss of 100 bytes or so is acceptable.

03 Listen to a commercial FM radio station

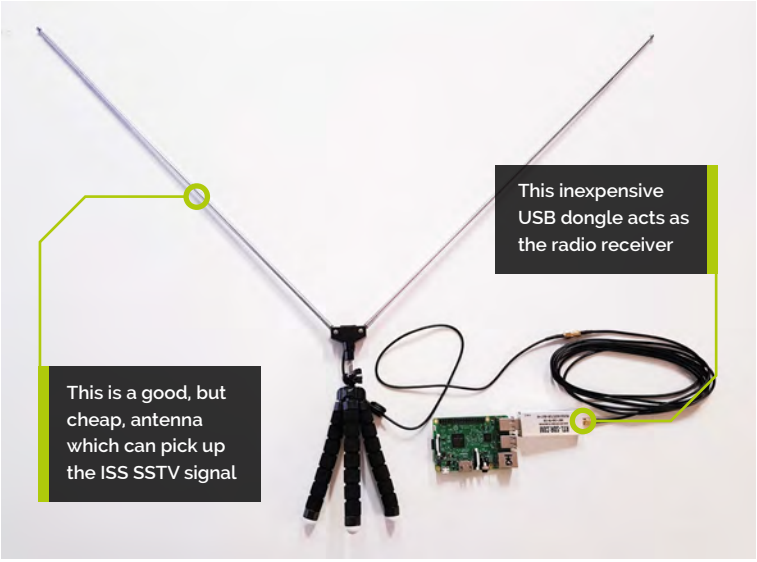
Let's tune into a commercial FM radio station to check that both the dongle and the antenna are working correctly. The command below pipes raw data from rtl_fm into 'play', which will then produce the audio output. Modify the value after the `-f` to specify your own FM station/frequency and don't forget to put the capital M after the number so it knows you mean megahertz and not hertz (98.8MHz is BBC Radio 1 in the UK). In a Terminal, enter:

```
rtl_fm -M wbfm -f 98.8M | play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```

Note that the 'Tuned to' value in the output doesn't match. This is due to a property of all SDR hardware, called 'DC spike', and can be ignored. Plug in your speakers/headphones and you should now be able to hear audio from the commercial FM station. Move the antenna around or select a different FM radio station with a closer transmitter if the audio is noisy. Press **CTRL+C** to quit from rtl_fm.

04 Set up QSSTV

QSSTV is the program that will listen to the audio output on the Raspberry Pi and decode the pictures from the fax/modem noise when the


This inexpensive USB dongle acts as the radio receiver
This is a good, but cheap, antenna which can pick up the ISS SSTV signal

ISS is in range. Let's just check the settings are all correct. In the main menu, select Internet > QSSTV, then select Options > Configuration and choose the following options in the Sound tab:

Audio Interface – PulseAudio
Input and Output Audio Device – default -- Playback/recording through the PulseAudio sound server
Sound Input – From sound card
Sound Output – To sound card

Click OK. Next, run the Calibrate function and go make a cup of tea/coffee: Options > Calibrate. Click OK when both progress bars reach 99%.

05 Decode a picture from test audio

Now that QSSTV is set up, we can download a test audio file, play it, and see a picture slowly appear. In a Terminal, enter:

```
wget https://raw.githubusercontent.com/davidhoness/sstv_decoder/master/sstv_test.mp3
```

Select the Receive tab in QSSTV and choose the following settings:

Use VIS – on
Auto Slant – on
Autosave – on
Signals – Normal
Mode – Auto

Top Tip

Play with SSTV on mobile

For a quick test, you can try these free mobile apps: Robot36 on Android; CQ SSTV on iOS. Play this MP3 (**magpi.cc/ruANWC**) and put the phone near the speaker.

Top Tip

RTL-SDR starter kit

Your author personally recommends this starter kit that comes with the USB dongle and a good v-dipole antenna: **magpi.cc/uBMKQg**

Click the play ▶ button. Nothing will happen, but just click it to start the receiver.

Open **sstv_test.mp3** in the Chromium browser. You can either drag and drop the file from File Manager, or enter **file:///home/pi/sstv_test.mp3** into the address bar.

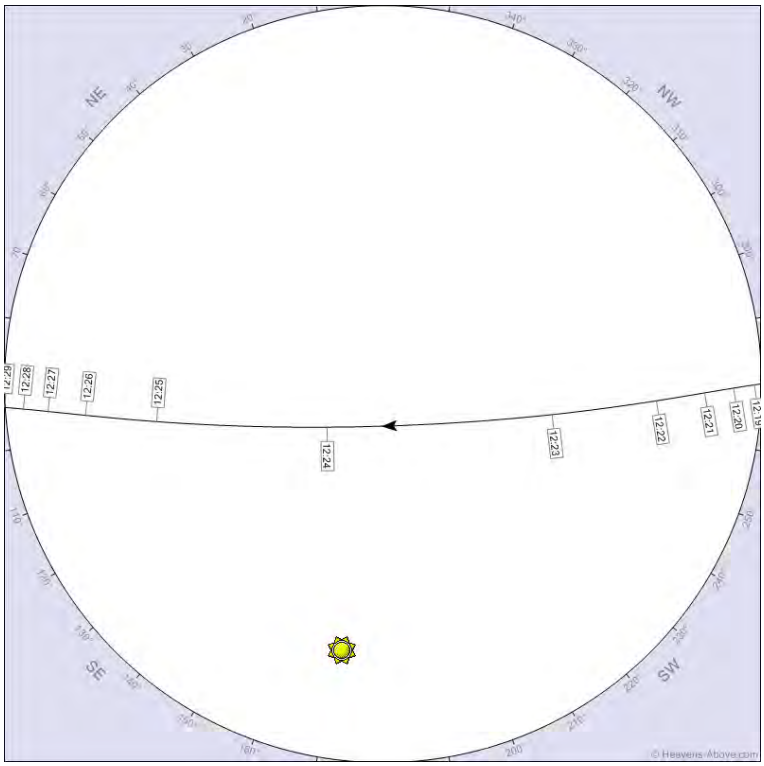
Let it play and watch the picture appear in QSSTV. Notice the waveform and waterfall display activity on the right of the screen. Other SSTV test recordings can be found online, such as: **magpi.cc/dXpjRr**.

06 Prepare for the next SSTV event

To get pictures from the ISS you need to know three things:

- When the ISS will be transmitting the SSTV signal. It's not always-on.
- What frequency the SSTV signal will be transmitted on (usually 145.8MHz).
- When the ISS will fly over your location.

The ISS flies over most places a few times every day (unless you live in the Arctic/Antarctic). You can find out when the ISS is transmitting on this website: **ariss-sstv.blogspot.com**.



Imagine holding this picture above your head and aligning it with the compass directions. The line shows the flight path of the ISS across the sky above you

Note that there is an SSTV event starting on **12 April 2019** celebrating the launches of Yuri Gagarin in 1961 and the first Space Shuttle, 20 years later, in 1981. These events usually last for a few days at a time, meaning you will get multiple ISS passes during which you can receive the SSTV signal.

07 When is the ISS flying over me?

There are many websites you can use for this, but your author's personal favourite is **heavens-above.com**.

Click 'Unspecified' in the top right to set your location. Type your postal code and country into 'Enter place to search for' and click Search. Scroll down and click Update. You'll now be back on the homepage; under Satellites, click ISS. For 'Passes to include', select 'all'.

The table now showing lists the times of the upcoming ISS passes for your location. The Alt value under 'Highest point' is the maximum elevation of the pass above the horizon; 90 would mean it goes directly overhead. Passes with higher maximum elevation will give better results than lower ones. Clicking on any row in the table shows the sky chart for that pass. Imagine holding that picture above your head and aligning it with the compass directions. The line shows the flight path of the ISS and this is useful for roughly positioning your antenna. It can also be useful to look at the 'Ground track' (see link in top right). Plan ahead and set up your ground station with time to spare before the ISS flies over.

08 Adjust your antenna

If you are using the RTL-SDR kit antenna, please read the getting started guide for it: **magpi.cc/qzUKtZ**.

It's a good idea to adjust the length of the telescopic arms based on this dipole calculator (**magpi.cc/uGNNxP**) to match the antenna's resonant frequency to 145.8MHz. This will help pick up weaker signals from low elevation passes.

Under Desired Frequency, enter 145.8. Under Select Antenna Calculation, select One Side. Click Calculate. The answer should be 48.9cm.

Remember, there is about 2cm of metal in each side of the antenna base. So take 48.9 and, before screwing in the arms, use a tape measure to make them both this length minus 2cm (excluding the

screw threads). It doesn't have to be perfect, but make it as close as possible.

09 Deploy your antenna

As a general rule, try to get your antenna up as high as possible with a good view of the sky in all directions. Outside is always better, but you can still get the signal from inside a house – you'll just see more noise in the pictures. Then, with the dipole arms flat, roughly align them with the flight path of the ISS that you find on **heavens-above.com** and bend the two arms upwards into a right-angle V-shape. You don't need to realign it for every pass, but it will yield better results if you do.

Antennas and their performance are sometimes a bit of an art form as opposed to an exact science. Experiment with different configurations on different ISS passes. Try one arm horizontal and one arm vertical, for instance.

10 Set the correct time again

QSSTV saves the decoded images with a UTC time file name and this is useful later when working out which ISS passes they are from. In a Terminal, enter:

```
sudo ntpdate pool.ntp.org
```

11 Tune into the ISS frequency

We'll use rtl_fm again for this, but note that the command is different to the commercial radio station one. A commercial radio station uses *wide band* FM whereas the ISS SSTV transmission uses *narrow band* FM and so we have to set rtl_fm up differently. In a Terminal, enter:

```
rtl_fm -M fm -f 145.8M -s 48k | play -r
48k -t raw -e s -b 16 -c 1 -V1 –
```

You should be able to hear white noise now (don't forget the capital M for megahertz), but this will change to the fax/modem noise when the ISS flies over with the transmission active. Keep this window open. A picture is transmitted from the ISS continuously for about two minutes, with another two minutes of off-time before the next picture starts.

▲ The ARISS VHF antenna installed on the bottom of the Columbus module *Credit: NASA/ESA*

12 Start QSSTV

In the Raspbian main menu, select Internet > QSSTV. Select the Receive tab in QSSTV and choose the following settings:

Use VIS – on
Auto Slant – on
Autosave – on
Mode – Auto
Save if Complete (%) – 10

Click the play ► button. The waveform and waterfall display on the right should show noise coming from rtl_fm. Leave QSSTV running.

> "Antennas and their performance are sometimes a bit of an art form as opposed to an exact science"

13 What is Doppler shift?

A common example of Doppler shift is the change of pitch heard when a police car or ambulance speeds past you. Compared to the emitted frequency of the siren, the frequency you hear is higher during the approach, identical at the instant of passing by, and lower during departure. The same thing happens with radio waves as with sound waves. The ISS is moving at 7 km per second!

To compensate for this, ground stations must continually retune their receiver as the ISS approaches (starting ~3.5kHz above 145.8MHz), passes overhead (reaching 145.8MHz exactly), and flies away (ending ~3.5kHz below 145.8MHz). The good news is that QSSTV is clever and can compensate by tracking the centre frequency of the audio signal as the ISS flies over. So you don't need to do anything!

Top Tip

Antennas to avoid

Avoid the little 30 cm mag-mount antennas that often come with these RTL-SDR dongles. You will have little or no hope of picking up the ISS with them.

▲ You can potentially get a certificate like this, even if you only decode a few lines of one picture

14 Compensate for Doppler shift (optional)

It doesn't make much difference, but if you want to have some fun with this, your author has created a Python script to compensate for Doppler shift. It tracks the ISS using ephem, computes the corrected frequency for when the ISS is passing over, and retunes rtl_fm using a UDP socket: **magpi.cc/zNtRDB**.

To use it you'll first need to run `sudo pip3 install ephem` and then just start the script in another Terminal window. But first you will need to modify the code to set your location. See the `LATITUDE`, `LONGITUDE`, and `ALTITUDE` variables at the top. Get the values from the top right of **heavens-above.com** and just estimate your altitude.

15 Wait for the ISS to arrive

You may want to leave your ground station overnight or over the weekend to capture several passes. When you return, select the Gallery tab in QSSTV to see what images were received. Alternatively, it's always exciting to watch the pictures coming in live as the ISS is above you!

16 Get an SSTV award!

On your Raspberry Pi, the decoded pictures can be found in **/home/pi/qsstv/rx_sstv**. You can upload them on this website for verification and recognition: **magpi.cc/Kdjnmy**.

Check the information for the event on **ariss-sstv.blogspot.com** to find out how to apply for the award certificates. This will be a PNG file that is emailed to you and will show a unique number. Good luck!

doppler.py

▶ Language: **Python**

```
001. #!/usr/bin/python3
002. import urllib.request
003. import math
004. import socket
005. import ephem
006. import time
007. import sys
008. import ssl
009.
010.
011. C = 300000000.0
012. F0 = 145800000.0
013.
014. LATITUDE = "52.219308"
015. LONGITUDE = "4.419926"
016. ALTITUDE = 20
017.
018.
019. class tle_reader(object):
020.     """
021.     For keeping ephem two line element sets up to date
022.     """
023.     def __init__(self,
024.                  tle_name="ISS (ZARYA)",
025.                  tle_file=
    "https://celestrak.com/NORAD/elements/stations.txt",
026.                  tle_max_age=3600):
027.         self._tle_name = tle_name
028.         self._tle_file = tle_file
029.         self._tle_max_age = tle_max_age
030.         self._tle = None
031.         self.reload()
032.
033.     def build_index(self, tle_lines):
034.         index = {}
035.         for i in range(0, len(tle_lines), 3):
036.             index[tle_lines[i].strip()] = (tle_lines[i +
    1], tle_lines[i + 2])
037.         return index
038.
039.     def reload(self):
040.         print("Loading: %s" % self._tle_file)
041.
042.         try:
043.             ctx = ssl.create_default_context()
044.             ctx.check_hostname = False
045.             ctx.verify_mode = ssl.CERT_NONE
046.             with urllib.request.urlopen(self._tle_file,
    context=ctx) as response:
```

DOWNLOAD THE FULL CODE:
magpi.cc/zNtRDB

```
                tle_lines = response.read().decode("utf-8").splitlines()
            index = self.build_index(tle_lines)
            tle_data = index[self._tle_name]
            self._tle = ephem.readtle(self._tle_name, tle_data[0], tle_data[1])
        except Exception as e:
            print(e)

        self._tle_age = time.time()

    @property
    def tle(self):
        return self._tle

    @property
    def tle_expired(self):
        return time.time() - self._tle_age > self._tle_max_age


class rtl_fm_remote(object):
    """
    For remote control of rtl_fm command line program
    """
    def __init__(self,
                 host="localhost",
                 port=6020):
        self._host = host
        self._port = port
        self._s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self._s.connect((self._host, self._port))

    def set_freq(self, freq):
        self.send_cmd(0, freq)

    def set_mode(self, mode):
        self.send_cmd(1, mode)

    def set_squelch(self, squelch):
        self.send_cmd(2, squelch)

    def set_gain(self, gain):
        self.send_cmd(3, gain)

    def send_cmd(self, cmd, param):
        cmd_bytes = (cmd).to_bytes(1, "little")
        param_bytes = (param).to_bytes(32, "little")
        self._s.send(cmd_bytes + param_bytes)

    def __del__(self):
        self._s.close()


rtl = rtl_fm_remote()
iss = tle_reader(tle_name="ISS (ZARYA)", tle_max_age=5520)

if iss.tle is None:
    sys.exit(0)

myloc = ephem.Observer()
myloc.lon = LONGITUDE
myloc.lat = LATITUDE
myloc.elevation = ALTITUDE

freq = F0
running = True

try:
    while running:
        myloc.date = time.strftime('%Y/%m/%d %H:%M:%S', time.gmtime())
        iss.tle.compute(myloc)
        alt = math.degrees(iss.tle.alt)

        if alt > 0:  # iss is flying over location
            new_freq = int(F0 - iss.tle.range_velocity * F0 / C)  # doppler
            if new_freq != freq:
                print(new_freq, round(alt, 2), myloc.date)
                rtl.set_freq(new_freq)  # set new frequency in rtl_fm
            freq = new_freq
        elif iss.tle_expired:
            iss.reload()  # we could be running for days / weeks
        else:
            time.sleep(10)  # do nothing, wait for iss to arrive
            freq = F0
except KeyboardInterrupt:
    running = False

print("Bye")
```

AMAZING HAM RADIO PROJECTS

Raspberry Pi and ham radio – a match made in amateur heaven. By **Rob Zwetsloot**

If you're a regular reader of *The MagPi*, you may have noticed a survey in your email inbox a while ago about the magazine. One of the questions asked was about the kind of topics you'd like to see in the magazine, and we got an overwhelming number of requests to cover ham radio.

Ever since the Pi came out, the ham radio community have been using it for fun and sometimes practical projects, so it's about time we let you know how your Pi can do some fun stuff with ham.

WHAT IS HAM RADIO?

Amateur radio has a long history. What makes it different to regular radio?

Ever since radio has been used to communicate, there have been amateur radio enthusiasts. As radio became increasingly widespread in the early decades of the 20th century, so too did amateur radio usage. As the number of enthusiasts grew, so did the chances for malicious amateur radio users, which eventually resulted in laws around the world limiting the frequency that these 'ham radios' could be operated under.

Over the years, the frequencies have changed, and new registration and licensing rules have come into place to make sure the amateur radio community knows how to properly enjoy their hobby without disrupting anything important.

Modern era

With the recent boom in smart devices, microcomputers, and other hobby electronics, the possibilities and accessibility of ham radio have only increased. Thanks to cheaper parts and more power to play with, we've seen plenty of Pi projects over the years that make use of ham radio. Hopefully we'll get to show you something you've not seen before over the next few pages, or give you inspiration for your own ham project.

Why ham?

In the early days of telegraphs, people who weren't so great at using Morse code were often referred to as 'ham' as an insult. Think ham-fisted, or how an actor 'hams it up'. Much like a lot of old pejorative terms, it's been reclaimed by the amateur radio community – and that's how we get to ham radio.

▲ A lot of army personnel had to use ham radios to communicate with loved ones during the mid-20th century

▲ Post Office engineers inspect radio equipment in 1897. *Credit: Cardiff Council Flat Holm Project*

◀ A German amateur radio enthusiast takes to the slopes with his 'portable' equipment

USING HAM RADIO WITH RASPBERRY PI

What you need to get started using amateur radio applications

As the Raspberry Pi has been so popular with ham radio operators for so long, there's a ton of software and resources readily available for folks who want to dabble with Pi-powered radio. As Raspbian is Linux-based, it also means a lot of existing packages have been ported to the Pi.

What you need...

ANY RASPBERRY PI – The newer the better, but a lot of applications will work on a Pi Zero as well as Raspberry Pi 3B+.

RASPBIAN – Most of the relevant software will be compatible with Raspbian, so if you're not running it, it's a good idea to prepare an SD card that does.

AMATEUR RADIO LICENCE – It's against the law in most countries to operate a ham radio without the relevant broadcast licences.

▼ A Raspberry Pi 3 Model B+ is recommended, although a Pi Zero can handle many radio applications

WARNING!

You need a licence to operate amateur radio in most countries. There are a lot of rules and regulations you need to follow, otherwise you may cause problems for others. To take a look at how to get a licence in the UK, head to the Ofcom website here: **magpi.cc/SivDkd**

Basic usage

Radio waves are one of the earliest used ways of encoding signals. Much how fibre optic uses light patterns to transmit data, and copper wire uses electrical pulses for data, radio waves are modulated in specific ways to hold data. The most common form is audio, and specific types of modulation (amplitude and frequency) are better for different purposes.

In the context of using amateur radio with Raspberry Pi, you use it to encode and decode radio waves in the same way dedicated hardware can. Using a full computer for this allows for more flexibility in the way you read and send data – for example, with the radio in your car you wouldn't be able to visualise aircraft patterns in the sky above you.

Ham Pi benefits

The Raspberry Pi is a great computer to use in conjunction with amateur radio due to its hackability, low-power requirements, and portability. You can easily read and even output relevant data via GPIO pins with the correct software, and carry around fully functional devices that will last a fair old time on a portable battery or similar.

Over the next few pages, we'll show you some of the things you can do with a Raspberry Pi and ham radio.

AMAZING HAM RADIO PROJECTS

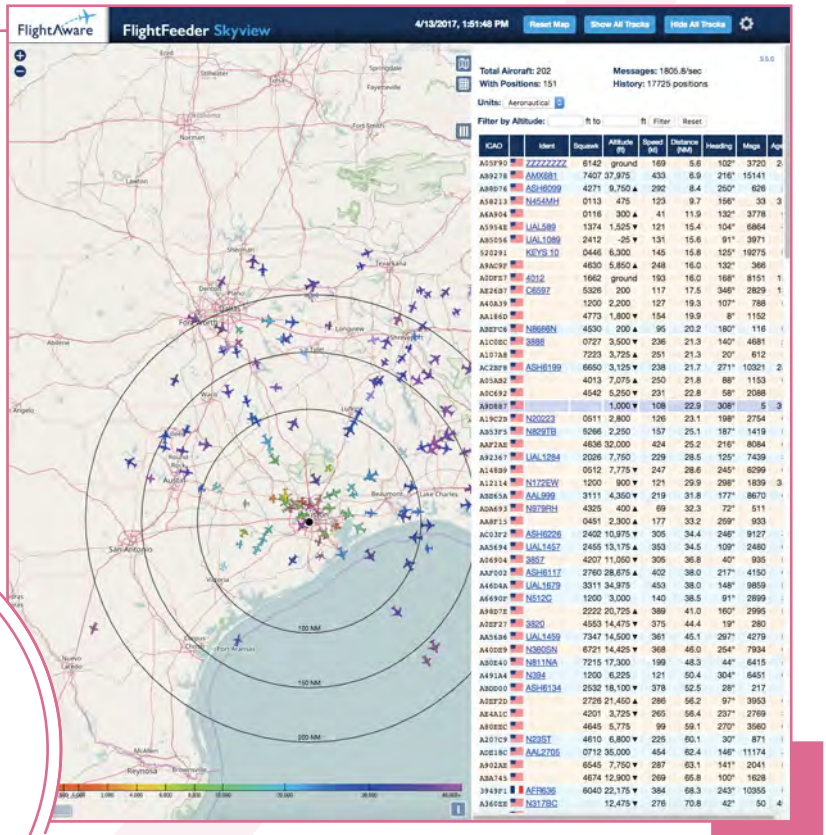
Not sure where to start? Take a look at these amazing guides!

ADS-B flight tracker

This is one of the more common ham radio projects you see that interacts with Raspberry Pi: turning your Pi into an ADS-B ground station and allowing you to track flights. Similar tech is used in high-altitude ballooning projects to track where the balloon is.

It works on line-of-sight range and can have a range of up to 300 miles. Using the radio data and FlightAware's PiAware software, you can then plot aircraft near where you are.

magpi.cc/phwYXr



▲ Use PiAware to turn your Raspberry Pi into an ADS-B ground station to track plane flights within a radius of up to 300 miles

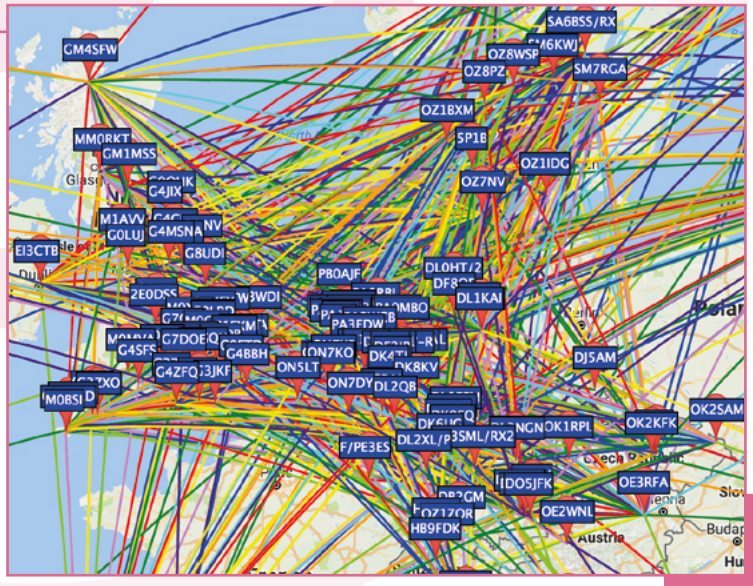
> "It's great for seeing if a friend or loved one's plane is delayed"

WSPR transmitter

WSPR (or 'whisper') beacons are quite common on a Raspberry Pi, allowing ham radio enthusiasts to create a network map of where folks are, and the range of their devices under specific frequencies and such so that they can figure out how useful those frequencies are. You can find the map and other details at **magpi.cc/uJnbUN**.

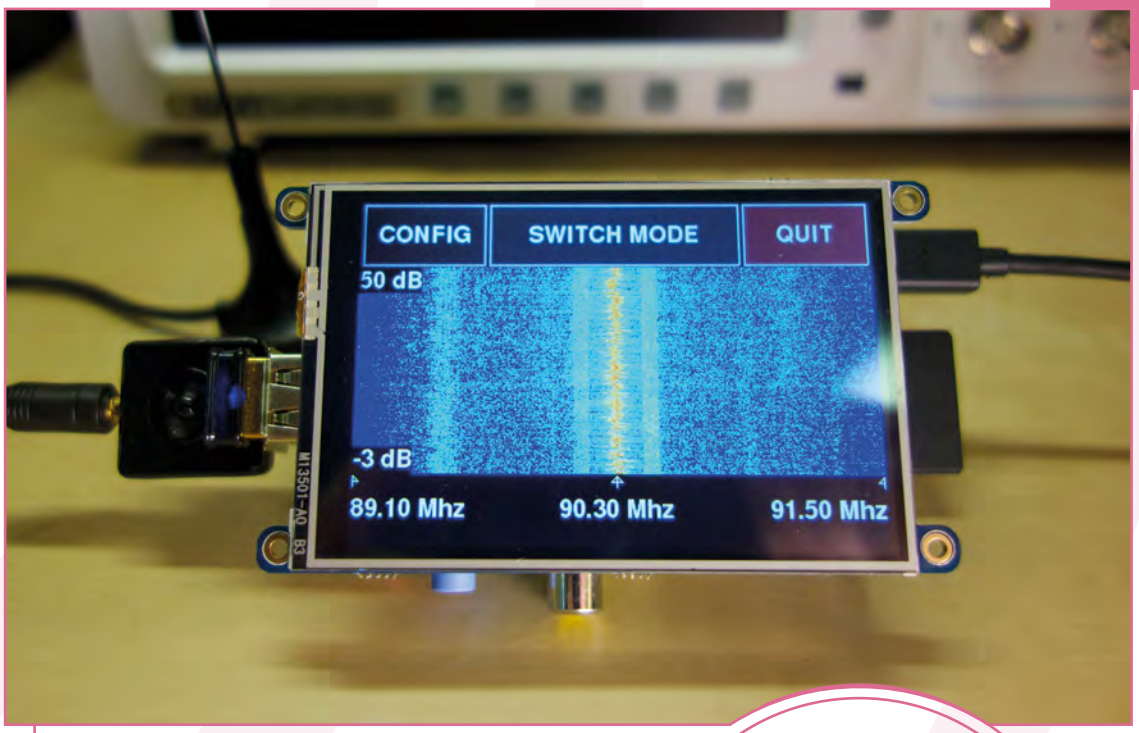
WSPR stands for 'weak signal propagation reporter', and makes use of low-power transmission to make its reports. There's software that uses it which is specifically made for the Raspberry Pi, called WsprryPi, and it works a little bit like the FM transmitter, requiring just an aerial to perform its function.

magpi.cc/EjQpbd



▲ Connect a couple of your Pi's GPIO pins, via a low-pass filter, to an antenna and you too can be on the WSPR network map

Also, don't forget to check out how to receive images from space using ham radio on p62!



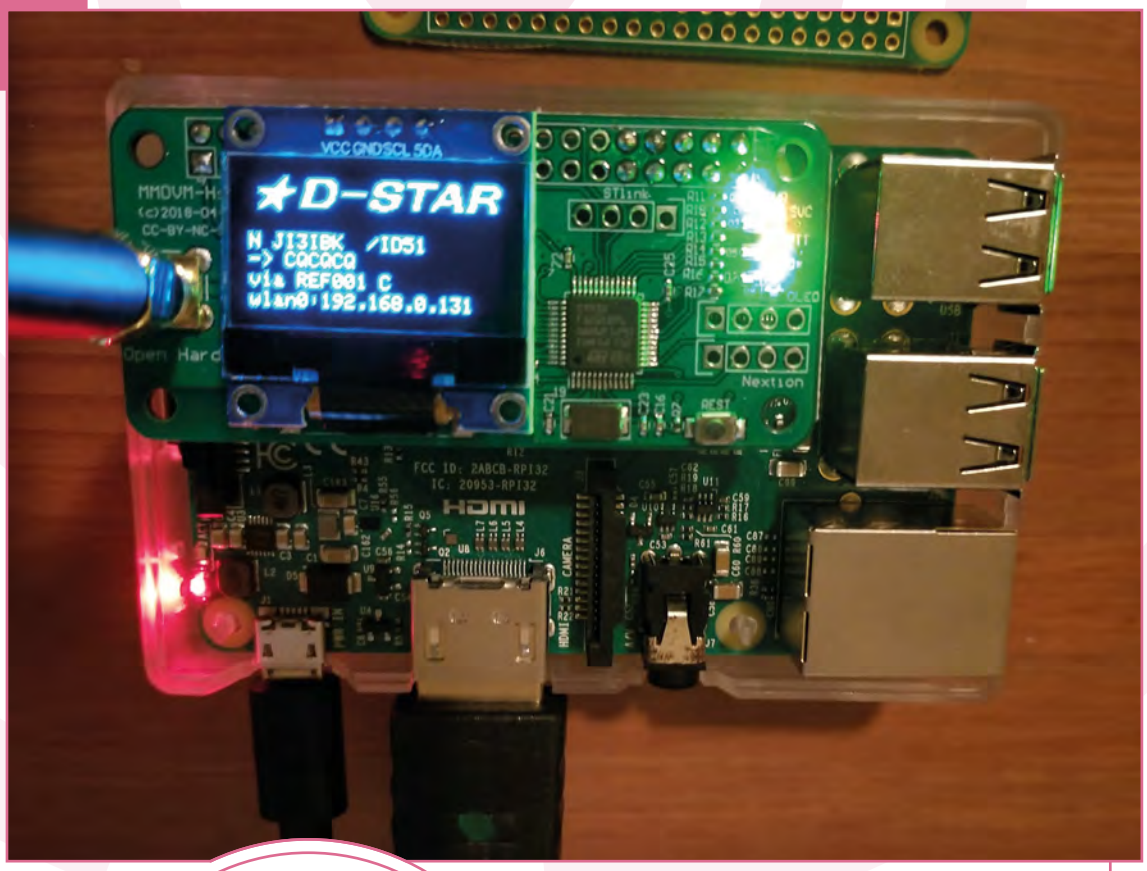
Remote SDR scanner

Software-defined radio, or SDR, is where the computer decides how to decode a radio signal, rather than any specific radio hardware attached to it. This adds a layer of flexibility to receiving a signal, and it's something you can create with a Raspberry Pi.

Although software-reliant, it does require the use of specific receiver hardware in the form of an RTL-SDR USB dongle to actually get the signals to then be processed. SDR# (read SDR sharp) is a popular piece of software for this, and the guide goes through how to set it up on Pi.

magpi.cc/jawZRE

> "SDR is where the computer decides how to decode a radio signal"



> It's able to relay digital transmissions across the network

Digital voice hotspot

Digital voice allows for the encoding and decoding of ham radio voice transmissions. Much in the same way that television went from analogue to digital, this has the benefits of improving the signal, while also using less bandwidth at the same time. There are a few specifications for this kind of usage, and this project makes use of D-STAR to handle it.

You'll again need specific hardware for this, and the jumper cable as an aerial trick won't work in this case. As well as working for your own transmissions, it's able to relay digital transmissions across the network.

magpi.cc/VjYZcp

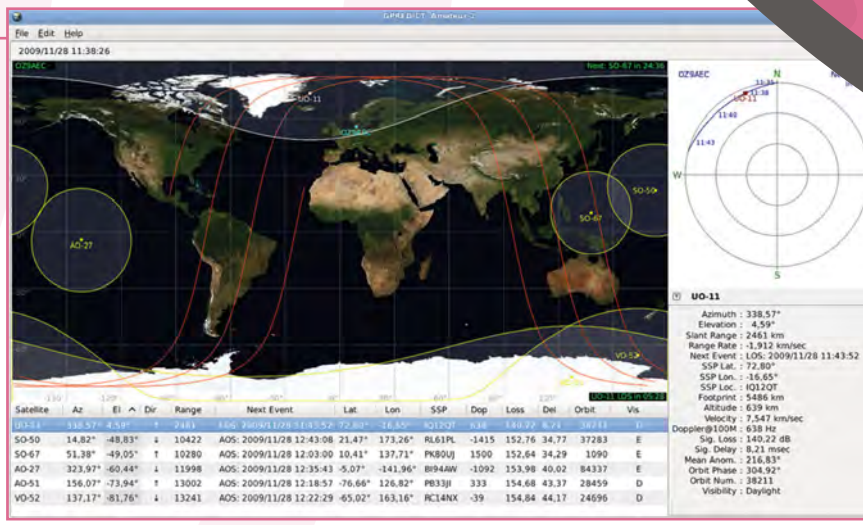
Got an amazing ham radio project of your own? Tweet us your photos! **@TheMagPi**

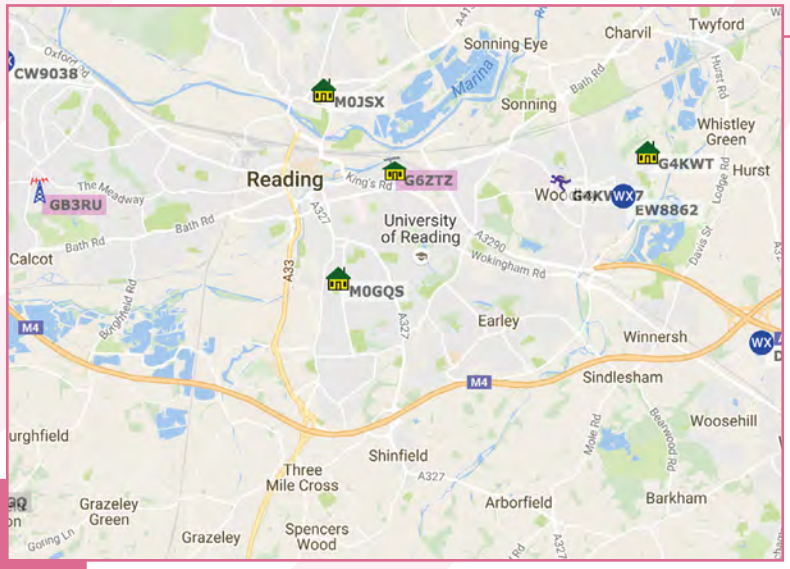
Satellite tracking

Think aircraft tracker but on a much grander scale – using the software Gpredict, you can track satellites. Up in space! That's very cool. It also includes great visualisation software that is highly customisable. We've seen this kind of software in the past being used to track the ISS – a good way to make sure you're out and ready if it passes overhead during the night!

It's also powerful enough to control antennas, if you have movable ones hooked up. You'll need to compile it on Raspberry Pi, though, so it's a much more advanced project.

magpi.cc/BqPWLm





▲ With a Raspberry Pi and an RTL-SDR dongle, you can set up your own APRS IGate

APRS IGate

By its very nature, some ham radio equipment isn't particularly powerful. Also, radio waves degrade due to interference and physics and such. What an Automatic Packet Reporting System (APRS) digital repeater does is allow packets from other radio operators to be digitally repeated online (Internet Gateway or IGate), so that they can travel much farther by then being transmitted somewhere else. It's also a good way to track yourself and other local ham radio folk.

Again, the setup is incredibly similar to an SDR, using the same dongle and software, albeit with some tweaks in how the data is handled.

magpi.cc/yppPRB

Breakout Garden HAT

SPECS

BREAKOUT SLOTS:
6 × 0.1" pitch, five-pin edge connectors

COMMUNICATION:
Shared I²C bus

EXTRA CONNECTIVITY:
20-pin breakout strip

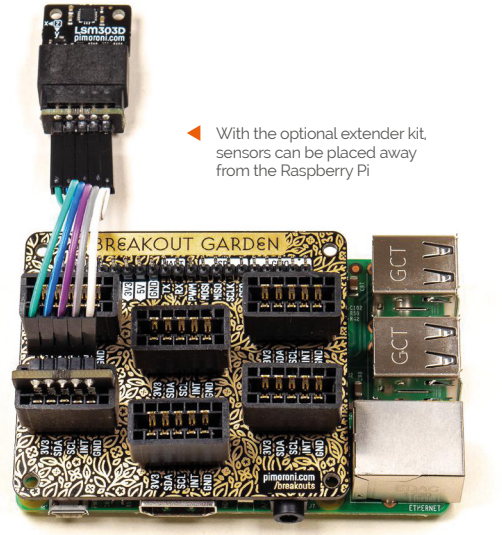
▶ Pimoroni ▶ **magpi.cc/AUCBrs** ▶ £13 / $14

With no soldering or wiring required, this offers an easier way to grow your projects. **Phil King** digs in

For those digital makers who want to get straight down to coding and creating, without having to first fiddle around with soldering and/or complex wiring, the Breakout Garden HAT should prove a real boon.

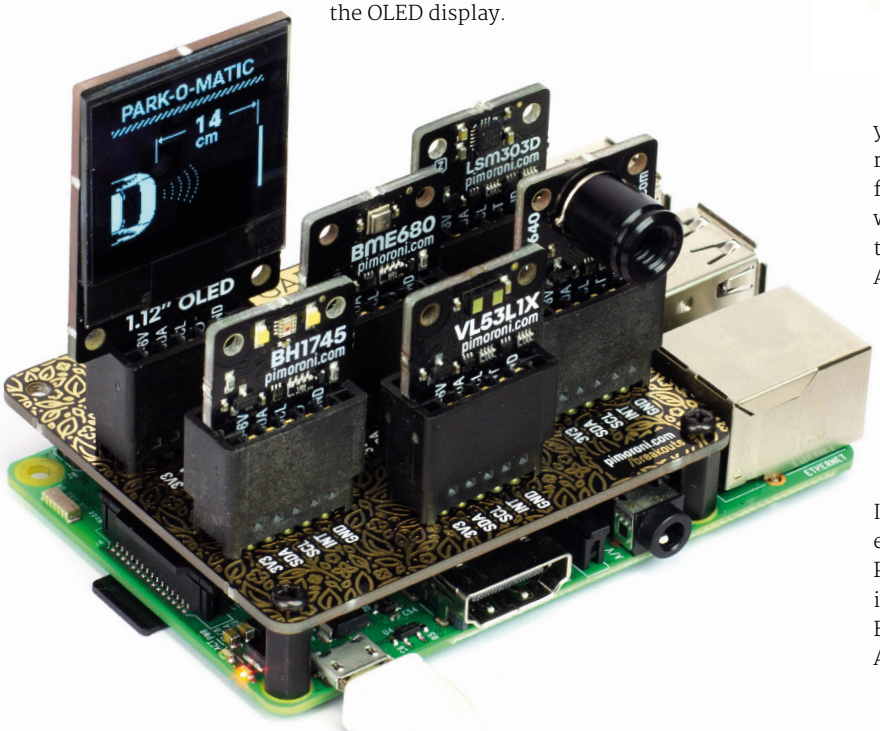
Akin to other 'connector' systems, such as Grove, it enables users to simply plug and play with various tiny breakout boards: at the time of writing, Pimoroni's range is 13 strong, including sensors, cute mini displays, and even a thermal camera (see the collection at **magpi.cc/TsRuUj**).

Just insert one or more boards into the six slots in the Breakout Garden HAT and you're ready to go. The mini breakouts feel secure enough in the edge-connector slots and are very unlikely to fall out, although there is a bit of back/front wobble if you touch them – particularly on the larger ones like the OLED display.



With the optional extender kit, sensors can be placed away from the Raspberry Pi

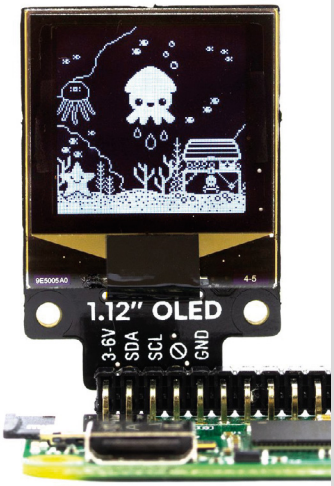
An HAT-ful of breakouts, including the OLED showing the car reversing sensor example



Reverse polarity protection is provided in case you accidentally insert a board the wrong way round. Also, it doesn't matter which slot you use for each breakout, as the latter's fixed I²C address will be recognised by the software, so it'll detect them correctly even if you move them around. And, since they use a shared I²C bus, this leaves plenty of spare GPIO pins, the most useful of which are broken out by a 20-pin strip of holes – a male header is supplied, which, due to the slightly offset alignment of the holes, fits firmly into place (or you could solder it).

Plant the software

Installing the necessary software is also easy enough. While, at the time of writing, the usual Pimoroni one-line curl installer wasn't working, it's easy enough to download and unzip the Breakout Garden GitHub repo (**magpi.cc/MvzTSO**). After inserting one or more breakouts into the HAT



▲ No soldering needed: just slot your mini breakout boards into any of the six slots on the HAT

▶ The 5x5 RGB matrix and mono OLED display breakouts

slots, you then just use a `sudo ./install.sh` command to automagically install the required software for them. The one exception to this was the 5×5 RGB matrix we tested, which required its library to be installed manually.

The GitHub repo features several Python examples to get you started. First, we tried out the weather project using the BME680 sensor and showing the results – including temperature and barometric pressure – on a neat 128×128 OLED mono display. Next, we used the LSM3030D 6DoF motion sensor with the OLED as a seismograph to detect 'dino stomps' – or, rather, us stamping on the floor! Another nice demo is using the laser-powered VL53L1X time of flight sensor as a mini car reversing indicator, showing the distance to the nearest object on the OLED. We also adapted this example to represent the distance with coloured squares on the 5×5 RGB matrix display.

Hanging baskets

While the HAT is great for prototyping software with mini breakouts, it's not necessarily ideal for a final working project – for instance, you might well want to reposition a distance sensor, or move a temperature sensor away from the Raspberry Pi for a more accurate reading.

> "Just insert one or more boards into the six slots in the Breakout Garden HAT and you're ready to go"

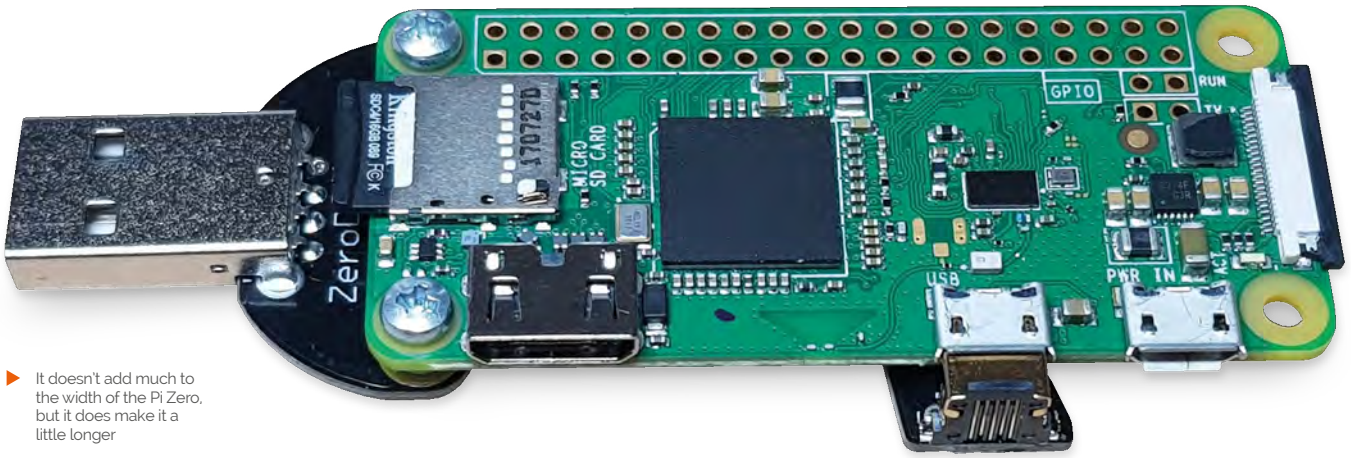
In this case, you could always solder pins (supplied along with right-angle headers) to the mini breakouts and connect them directly, or via a breadboard, to the Pi in a more permanent project. Alternatively, Pimoroni offers a Garden Extender Kit (**magpi.cc/xFojdG**) that enables you to extend the reach of your breakouts, so you could have them trailing like hanging baskets. Or, just using the Garden Extender parts and jumper cables, you could use any other I²C breakout with the HAT. So it's very versatile.

If you prefer a smaller form factor, a Pi Zero-size Breakout Garden pHAT is also available, although it only has three slots.

Verdict

Providing plug-and-play prototyping, this stylish and practical HAT is a great way to get creative without getting your hands dirty with soldering and wiring.

9/10



▶ It doesn't add much to the width of the Pi Zero, but it does make it a little longer

Zero Dongle

SPECS

USB PORTS:
1 × USB A,
1 × USB micro

MOUNTING:
2 × screws

DIMENSIONS:
70×35 mm

▶ 8086.net ▶ **magpi.cc/KHZHya** ▶ £4 / $5

Turn your Pi Zero into a USB gadget with ease thanks to this inexpensive add-on. **Rob Zwetsloot** gives it a try

One incredible use of the Pi Zero that often gets overlooked is its ability to function as a USB gadget – you can connect it to a computer's USB port and have it act like a USB plug-and-play device. This can be as simple as having it power on, act as USB mass storage, or even share your computer's internet connection.

The trickiest part of setting up a Pi Zero gadget is adding the necessary hardware to connect it. Male-to-male micro USB to USB A cables aren't very common, and not particularly neat. This is where something like the Zero Dongle comes in – slotting neatly under a Pi Zero, it's a solid piece of PCB connecting a USB A male connector to a micro USB male connector.

Dongle not dangle

The Zero Dongle fastens onto a Pi Zero (any Pi Zero) using a couple of screws on the SD card end. It's quite sturdy so you shouldn't have to worry about it snapping down the middle. There's only the smallest protrusion on the side, so it can slide into the Pi Zero's micro USB data port, although it does extend out a fair bit with the main USB A connector – but that's unavoidable.

▶ The thin bit of PCB is fairly sturdy

The dongle does not automatically set up your Pi as a gadget – there are no specific chips on it that will do that – so you'll have to make the usual modifications to your SD card. You also might have a little trouble finding a case that fits it.

Those are nitpicks, though – for the price and intended function, it's exactly what you'd need for a USB gadget add-on.

Verdict

A great and cheap add-on for a Pi Zero that allows it to work really well as a USB gadget device.

9/10

THE OFFICIAL Raspberry Pi Beginner's Guide

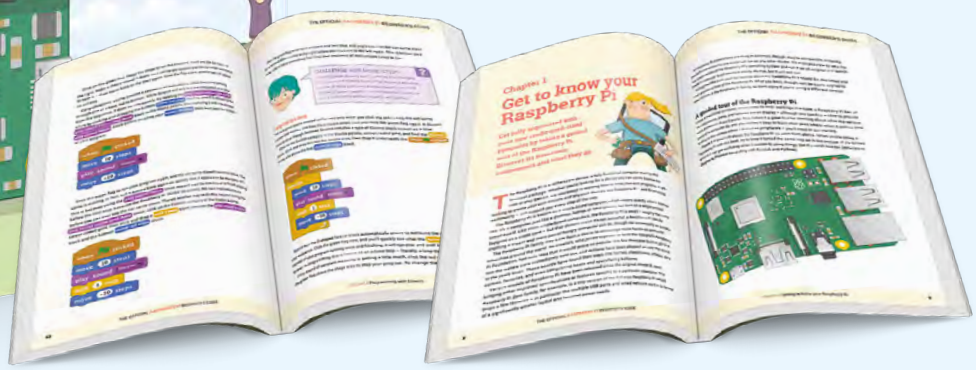
The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up the Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch and Python languages
- Create amazing projects by connecting electronic components to the Pi's GPIO pins

Plus much, much more!

£10 with FREE worldwide delivery

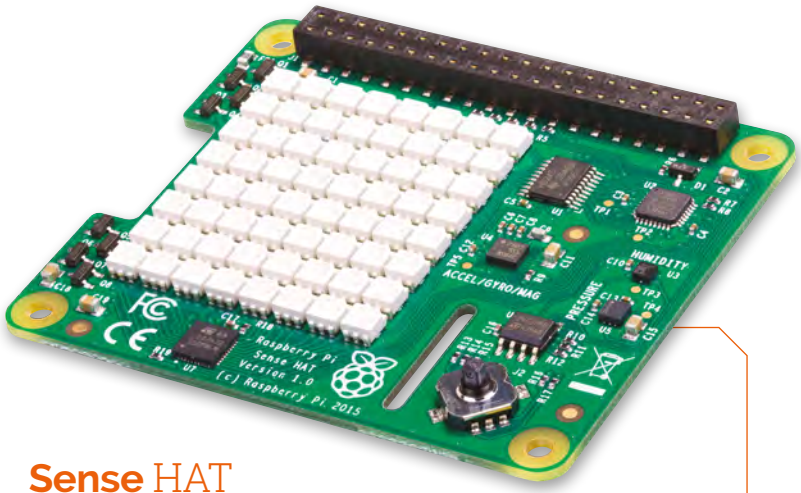


Buy online: magpi.cc/BGbook

10 Best: Raspberry Pi HATs

The very best Hardware Attached on Top for your Raspberry Pi

HATs are incredible add-ons to the Raspberry Pi that increase its functionality in a huge number of ways – from added sensors and inputs for fun projects, to practical applications in business and enterprise. Here are some of the best...



Sense HAT

Space-faring sensor

Used in the Astro Pi devices up on the International Space Station, this cool HAT has an 8×8 pixel display, environmental sensors, accelerometer, and a little joystick. There's loads of great Raspberry Pi resources that use it as well.

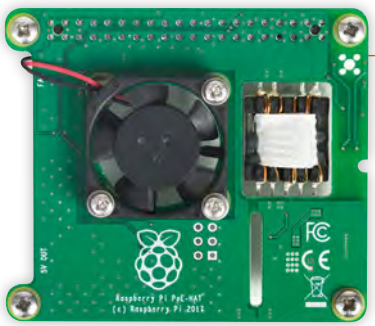
- £32 / $40
- magpi.cc/BsVbhG

GFX HAT

Display and inputs

We really like the GFX HAT – not only is it a very nice display that is easy to program for, it also has a series of (capacitive touch) buttons. It's amazing for practical projects or basically any Pi that's turned on a lot.

- £22 / $23
- magpi.cc/ZWvcLG



PoE HAT

Power over Ethernet

The Raspberry Pi is very popular in enterprise settings, and the PoE HAT allows for a more efficient way to add Pi boards to a system. With PXE boot and power from the Ethernet port, you can do a lot with a 3B+.

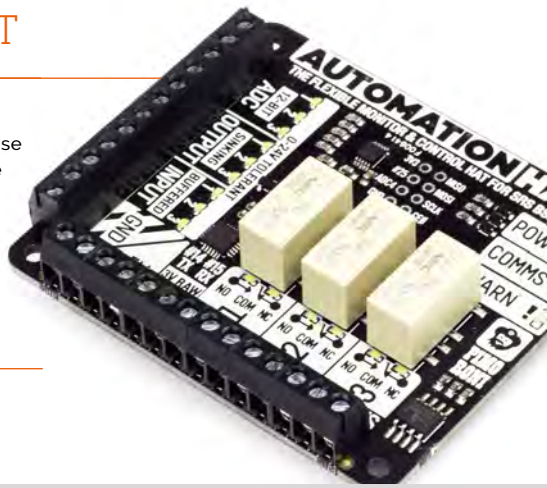
- £18 / $20
- magpi.cc/aqpwZc

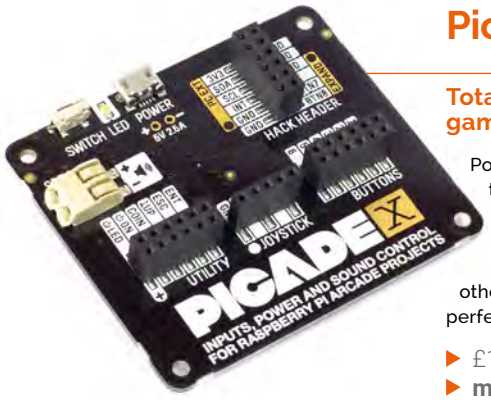
Automation HAT

Automating everything

Want to connect your entire house up to a Raspberry Pi? You'll have better luck with the Automation HAT, which has just a ton of connections that you can use. And it all sits on top of a Pi!

- £29 / $31
- magpi.cc/fgMGmr



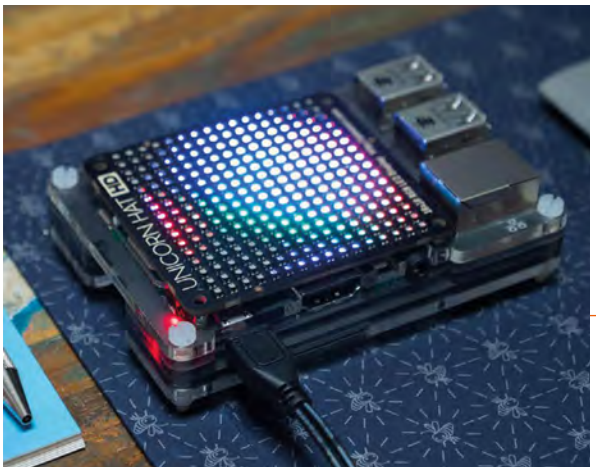


Picade X HAT

Totally awesome video games

Powering the almighty Picade is this amazing control HAT that's purpose-built for Pi-powered arcade machines. It includes audio and inputs, along with other bits and pieces, and it's the perfect heart for your arcade build.

- £16 / $17
- **magpi.cc/BupAFF**



Unicorn HAT HD

High-def LEDs

The wonderful Unicorn HAT HD lets you create amazing multicoloured visuals, perfect for a disco or project that needs a dot-matrix display aesthetic. It fits in a lot of Pi cases as well.

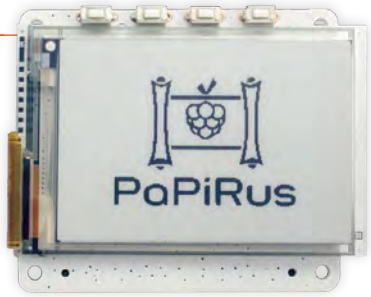
- £34 / $37
- **magpi.cc/uitfMn**

PaPiRus HAT

E-ink display

You can make anything look 20% classier with an e-ink display. Trust us, we've got the science to back it up. It's very low-power, depending on usage, and looks great even in bright sunlight.

- £41 / $54
- **magpi.cc/ikcQsi**

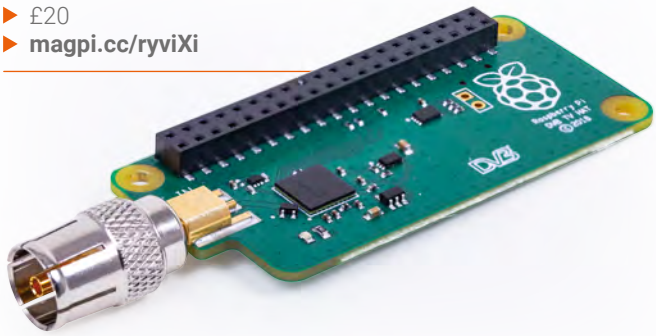


TV HAT

Must-see TV

Tune into digital TV in Europe with this excellent HAT which then lets you stream the signal around your house. It's the perfect complement to a Kodi box, and it sits nicely on top of a Pi Zero.

- £20
- **magpi.cc/ryviXi**

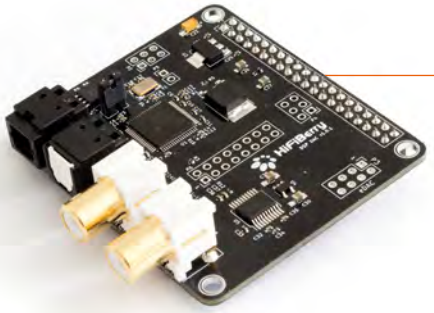


HiFiBerry DAC+ DSP

Powerful audio amplifier

This powerful DAC also includes a DSP (digital signal processor) which allows for full control of the way your audio is output from the Pi. The folks at HiFiBerry call it a 'DAC on steroids'.

- £60 / $78
- **magpi.cc/EsHtjE**



Flick HAT

Gesture control

The Flick HAT can handle all of your gesture control needs, with 3D tracking that lets you control the Pi with some simple gestures even from a moderate distance (about 15 cm away).

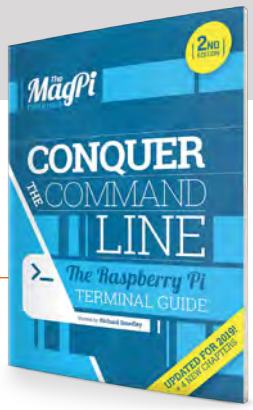
- £20 / $26
- **magpi.cc/vXZLzF**

HAT DEFINITION

Introduced in 2014 when the Raspberry Pi B+ was released, the HAT standard relies on the use of two dedicated GPIO pins on a 40-pin Raspberry Pi that allow for automatic configuration from the HAT so the Pi can use it. Learn more specifics here: **magpi.cc/PAHaGk**

Everything you need to learn The Command Line

Harness the power of the Raspberry Pi's command-line interface. **PJ Evans** gets typing



Conquer the Command Line

AUTHOR
Richard Smedley

Price:
£3.99 (free download)
magpi.cc/JrKoTA

Yes, we admit it, it's a bit cheeky to start this section with our own publication, but it wouldn't be right if we didn't feel we could recommend it! This newly revised book from Richard Smedley adds four new chapters to this already impressive guide.

This course is tailored around the Raspberry Pi experience, where others take a more 'generic' Linux approach. Squarely aimed at the absolute beginner, we start with how to 'find' the command line in Raspbian and gently up the pace, covering file handling, editing text, managing disks and networks, until finally touching upon more advanced topics such as processes and compiling software from source code. A light-hearted final chapter shows the fun side of the command line (there is one, honestly) where the student browses the internet in text-only mode, and it's not as pointless as you would think.

Throughout the guide, hints and tips grace the margins with shortcuts and technical explanations that even made this seasoned reviewer take a few notes. This is a great start for your command-line adventure, but not for those looking to deep-dive into shell scripting and other more advanced topics. Start here.

Websites

Great sites for online learning

CODECADEMY

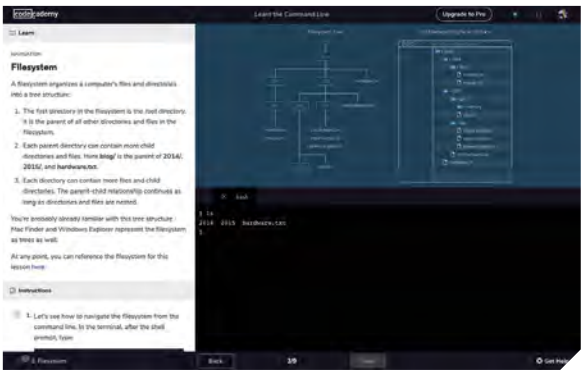
If you're unsure about using a 'live' prompt, Codecademy offers an on-line course with an interactive virtual terminal.
magpi.cc/VvTNvK

LINUXCOMMAND.ORG

If you've decided to tackle William Shotts's book, *The Linux Command Line*, this site acts as a great companion.
linuxcommand.org

TREEHOUSE

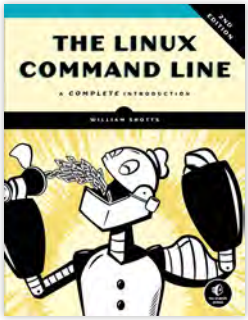
A subscription site that provides a wealth of training courses, including our friend the command line. Free trial available.
magpi.cc/AGHExX



The Linux Command Line

AUTHOR
William Shotts

Price:
£23.79 (free download)
magpi.cc/nvMNCo



So, you're serious about the command line. You want to be a master; a guru. You picture people spending days climbing a mountain just to meet you and ask about that obscure tar parameter. Well, you'd better get reading this book.

The Linux Command Line, now in its fifth edition, is as close to a definitive guide as you're likely to get. This is a real no-messing-about guide to controlling Linux systems, Raspberry Pi's Raspbian included. No stone is left unturned as we go from first principles through to advanced topics such as scripting, even including a chapter on many experts' favourite text editor vi, famed not only for its power but also its steep learning curve.

The content is well organised, almost academic in structure. This can look a little intimidating, but the author's writing style is friendly and accessible, avoiding arcane language. Especially impressive is the time taken to explain concepts and terms that others would (wrongly) assume the reader would understand.

Shotts, a firm believer in the open-source movement, has made the book available as a free download under a Creative Commons licence, but at 500-plus pages, you may well prefer the paperback from No Starch Press.

Command Line Kung Fu

AUTHOR
Jason Cannon

Price:
£8.99 (free download)
magpi.cc/GYQcBf

That innocent blinking cursor on the Terminal screen hides all manner of powers. It's only after some time and experience at that little prompt that the sheer scale of commands and capabilities dawns on you. Then, consider the thousands of applications that can be installed with APT. If you want to do something on the command line, chances are someone has already written it and you can install it on your Raspberry Pi in seconds.

Knowing the capabilities of all these applications is beyond most people's reach, and the same could be said for all the tricks and syntactic gymnastics that can be performed to automate workflows. Jason Cannon's book is not a tutorial, but instead a series of recipes to performing more advanced work in the Terminal. Like a stream of answers to difficult Stack Overflow questions (see 'Community' box), Cannon covers an eclectic range of subjects, from downloading webpages automatically to advanced text manipulation.

This is a reference guide rather than a course. It's a no-nonsense read but provides a ton of inspiration. A book that you can pick up, flick to a random page, and find yourself saying 'I never knew that'.

Community

Stuck? There are plenty of helpful folk out there.

RASPBERRY PI FORUMS

We're lucky to be part of a great community. No matter how seemingly trivial a question, someone here will help.
rpf.io/forums

STACK EXCHANGE

Huge question and answer site. In the unlikely event your question isn't already here, it's a very friendly place to ask.
stackexchange.com/sites

UBUNTU FORUMS

Raspbian is based on Debian, as is Ubuntu. The Ubuntu forums are a wealth of Pi-compatible information.
ubuntuforums.org

Claire Pollard

Maker, marketer, and more. We talk to Claire Pollard about using Raspberry Pi devices all day as a job

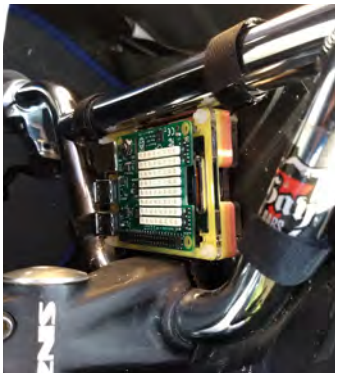
- Category Maker | Day job Technical Marketing Manager
- Website modmypi.com / piborg.com

When you've been around the Raspberry Pi community for a while, there are a few faces that start to become familiar. If you've ever visited the PiBorg desk at a Raspberry Jam, or gotten involved in the online Raspberry Pi community, you've likely bumped into Claire. Recently, she's become more active in the community, so we caught up with her to see what's up.

"I am a mathematician/computer scientist by training and have spent the last seven and a bit years working as a developer for a software company who make software for use in computer-aided design and computer-aided engineering," Claire explains. "My job needs a lot of different skills, from technical understanding of maths, to coding, to the ability to explain my work at different levels, from academic papers to social media.

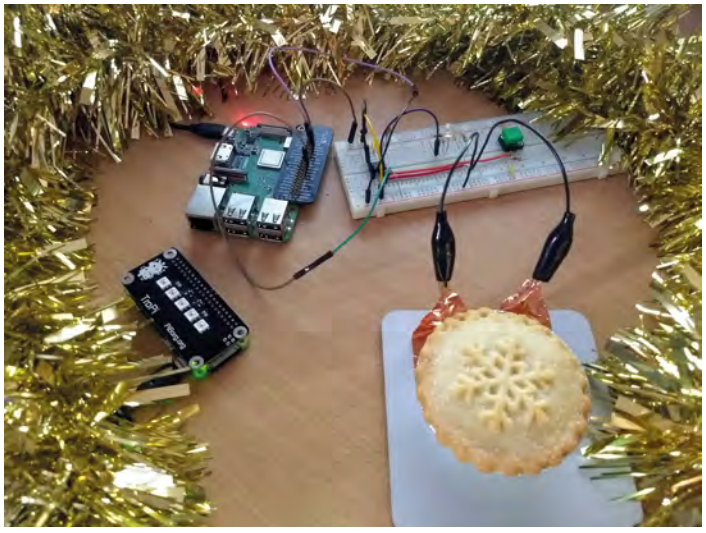
"Last year, I decided to leave that job and took up a position at ModMyPi to work in the Pi community full-time. The Raspberry Pi community has always been a part of my life as my fiancé Timothy owns PiBorg, so I've always been to shows, helped pack and assemble kits in the early days, and helped out where I can and really enjoyed the friendly, diverse, and supportive community. When the job came up at ModMyPi, I couldn't resist diving in and getting to use the Pi all day, every day!"

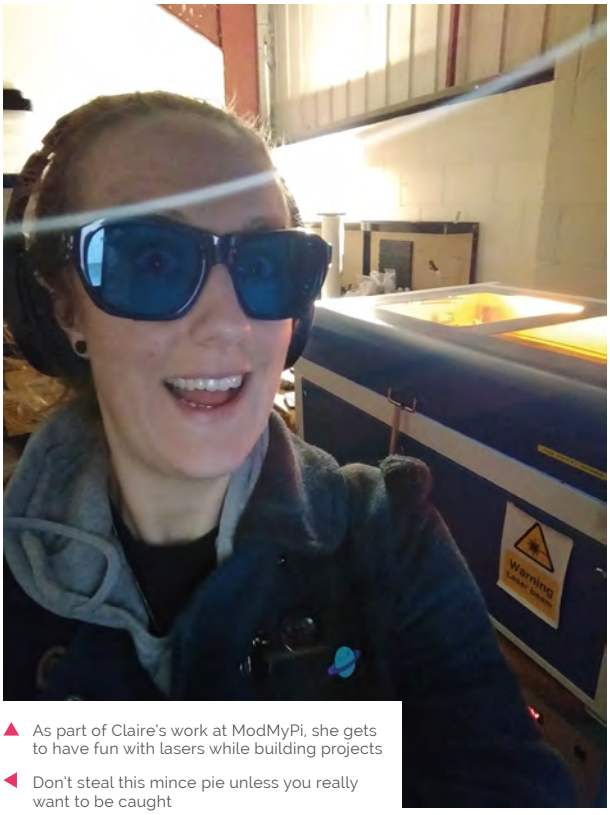
Claire is also the voice of Formula Pi, the amazing robot racing league set up by PiBorg.



▲ Claire combined her love of Raspberry Pi and BMX to create this ride tracker

◀ Wheeled robots line up for the start of a Formula Pi race, with Claire providing the commentary





▲ As part of Claire's work at ModMyPi, she gets to have fun with lasers while building projects

◀ Don't steal this mince pie unless you really want to be caught

When did you first learn about the Raspberry Pi?
My Dad is a big Linux fan, and he heard about the Raspberry Pi through one of the podcasts he listens to. He used to pick me up from work on his way home before I got a car, and we chatted about it on the drive home. He seemed quite excited at the prospect of a tiny Linux board that was easy to use and relatively cheap.

I think we missed getting one on release week, but Timothy got one on eBay a few weeks after launch, and the rest is history!

Do you have any favourite Pi projects you've made or been involved with?
I've really enjoyed helping with Formula Pi (**formulapi.com**) and seeing it grow into the chaotic event that it is from the initial Kickstarter we built a couple of years ago. I commentate on the YouTube feeds, which again is something completely different from my day job, as working from home I don't really get to talk to many people. It's like Formula Pi has become my little playground to let off steam and talk relentlessly about racing and other rubbish for an hour or two each week.

> "When the job came up at ModMyPi, I couldn't resist diving in and getting to use the Pi all day, every day"

The other project I'm enjoying is building a Sense HAT-based computer to go onto my BMX, to measure g-forces and accelerometer data from when I ride the track. I'm a British Cycling coach and ride BMX at our club, Royston Rockets. I used to race at an international level, but have since semi-retired due to injuries.

I love combining my passions, so this seemed like an interesting project where I got the opportunity to do just that. I'm almost at the point where I've got a working prototype, so testing should happen soon.

Finally, the Christmas Mince Spy project (**magpi.cc/aikrnL**) I completed was a lot of fun and I learnt about latch circuits whilst I built it!

Claire's maker tips

"Don't be afraid to just jump in and try something. Just having a go, not expecting to be the best straight away and not worrying about making mistakes is the best attitude to have if you want to get into making. Some of the things I've made in the past have been awful, but I've learnt so much from the mistakes and the almost working projects, it's helped me grow and become more confident with making!"

This Month in **Raspberry Pi**

#MonthOfMaking

The online maker community has stepped up to the #MonthOfMaking challenge

Last issue we announced #MonthOfMaking – our month-long event where we want the maker community to post their builds to Twitter using the #MonthOfMaking hashtag. We've seen loads of interpretations of what this means; here are some of our faves.

Dave Lauchlan
@davelauchlan
Follow
And from last night's photos we have several frames of this character, who visited us several times over the space of three hours.
#MonthOfMaking
02

01. Raspberry Pi community manager Alex has been having a lot of fun with the laser cutter recently – this frame is one of her many creations!

02. Hedgehogs are adorable, and so are these nature pics of the nocturnal creatures taken by a Raspberry Pi

03. Recycling tech with Pi is a popular pastime, including replacing an entry panel with an older screen

04. Making robots is very cool, especially when said robots win competitions

05. Another of Alex's laser-cut creations is this wonderful light-up Pi logo

06. The Raspberry Pi North America team got involved during their Maker Day

07. We very much believe that baking cakes can count as making, although we'd have to try one or two to make sure

08. Pi-hole is very useful, especially if you're a cybernetic life-form

03

Nick Reynolds
@ncmreynolds
Follow

Messing around fitting a @Raspberry_Pi into an old door entry panel. Mostly to re-use the 10.1" wide-screen display. It's only driven by a composite input but looks good enough for what I need. #MonthOfMaking

9:58 PM - 8 Mar 2019

04

Otter Robotics
@OtterRobotics
Follow

We came #second at the @Rotary tech day. 🎉🎉🎉 This was the robot that we used. 🤖 We even got a trophy. 🏆 We had to make a robot that was capable of picking 🖐 up 'radioactive ☢ waste 🗑' and moving it back to the start line.

#Robot #Robotics #MonthOfMaking #Rotary

2:56 PM - 6 Mar 2019

05

Raspberry Pi
@Raspberry_Pi
Following

Here is a beautiful light-up Raspberry Pi box that Alex made for @TheMagPi's #MonthOfMaking. She shows you how to make one for yourself, too. raspberrypi.org/blog/light-ras ...

PM - 7 Mar 2019

06

Christina Foust
@Foustberrypi
Follow

It's maker day at @Raspberry_Pi and the Oakland team is trying out some projects. What are you making today? #MonthofMaking #WomenInSTEM

6:10 PM - 7 Mar 2019

07

Karen Mouws
@LimsKaren
Follow

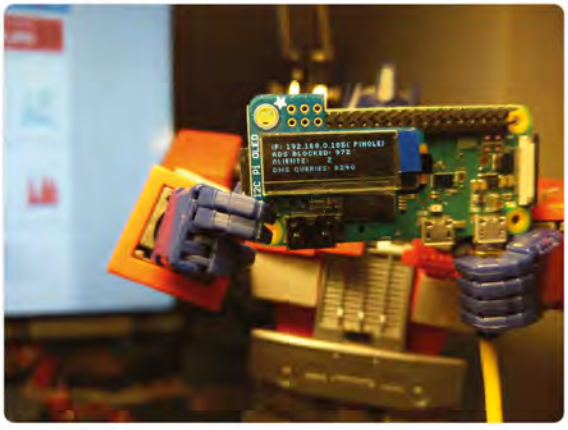
Does this #cakeinajar count for the #MonthOfMaking ?
Red velvet with chocolate mascarpone buttercream and hazelnut bits.

4:19 PM - 7 Mar 2019

08

Dermot Byrne
@dermotbyrne

@TheMagPi
Optimus Prime holds up a pi zero with the Ada fruit OLED installed running Pi-Hole ad blocker #MonthOfMaking
@JosephMcNally5



12:29 PM - 13 Mar 2019

Big Birthday Weekend 2019

The Raspberry Pi has turned seven, and people around the world have been celebrating

March isn't just the #MonthOfMaking, it's also when the Raspberry Pi usually gets its own birthday party! Like in 2018, a global series of jams were organised to celebrate the seventh year of Pi. Here's some of the fun...

01

02

04

SRVEF Imagineering
@srvef_Imagineer
Follow
Students across our district helped kick off the @Raspberry_Pi Big Birthday Weekend! Not a bad turn out for a bunch of beginners... 1st Birthday Jam for teachers and students! #rjam #PiParty Happy 7th Birthday #RaspberryPi !
1:10 AM - 3 Mar 2019

03

01. From Peru we have this wonderful custom laptop being shown off

02. R2-D2 decided to show up to a birthday Jam in Lebanon. Must have been that lovely cake

03. Students from across California celebrated the Raspberry Pi – a first for the teachers and students there!

04. Looks like this Raspberry Jam photo-booth needed a bit of testing

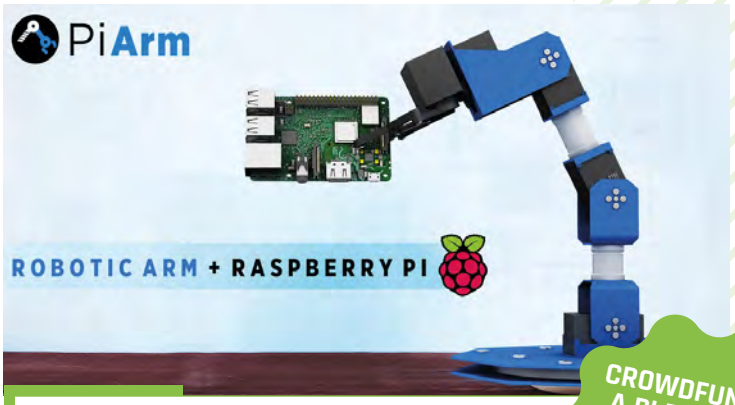
Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month

Strawberry4Pi

We've never seen a project where the Pi just slots into a board before (the Compute Module doesn't exactly count), and this one's function is to give ultimate IoT access to a Pi Zero. It uses its own software, and all you need to do is attach your automated appliances.

▶ **kck.st/2Svrqgm**



PiArm

A powerful DIY robotic arm from our friends at SB Components, this one has a lot more articulation than the MeArm. It's also controllable from many more sources (such as mobile phones as well as full automation), and both the hardware and software is open-source.

▶ **kck.st/2TCbUnG**

CROWDFUNDING A PI PROJECT?

If you've launched an irresistible Pi-related project, let us know!

magpi@raspberrypi.org

Best of the rest!

Here are some other great things we saw this month

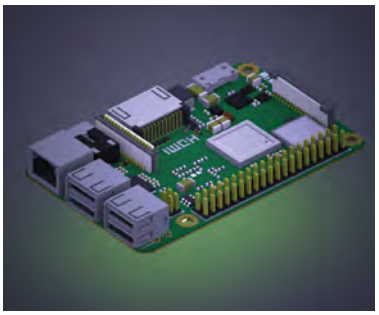
NES CLOCK

We love this custom-designed clock inspired by the NES. The controller isn't there just for show, either – it runs RetroPie as well as being a clock. That's the kind of alarm we can get behind.

▶ **magpi.cc/wkWiac**

VOXEL RASPBERRY PI

This recreation of the Raspberry Pi 3B+ made out of voxels (think 3D pixels) is great. And apparently anyone is welcome to use it! Be a good sport, though, and credit the creator, reddit user Cal_Macc.



▶ **magpi.cc/GQMgry**

PSPI

The PSP is a cult classic handheld and ever since it came out, people have been hacking and modding it. The next logical step was to install a Pi Zero inside one and have it power the screen. A great idea.

▶ **magpi.cc/RhZZTB**

03

Raspberry Jam Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. PiJam.cz

- Monday 1 April
- **Impact Hub Praha D10, Prague, Czech Republic**
- **pijam.cz**

One of three PiJam events running this week in the Czech Republic; the others are in Brno and Ostrava.

02. Komarovi-TSU Jam

- Saturday 6 April
- **1, Chavchavadze Ave, Tbilisi, Georgia**
- **magpi.cc/igbNVu**

This first Raspberry Jam in Georgia will introduce folks to physical computing with the Pi.

03. Santiago Raspberry Jam

- Friday 19 April
- **Corniel Academy, Santiago, Dominican Republic**
- **magpi.cc/wjvFqP**

We don't see many Jams in the Caribbean, so it will be well worth attending this first one!

04. Raspjamming

- Friday 26 April
- **Technischen Universität Graz, Graz, Austria**
- **magpi.cc/pvPMAL**

A place for people to meet to work on Pi projects – bring your own, or join in on the provided challenges.

05. April Exeter Raspberry Jam

- Saturday 6 April
- **Exeter Library, Exeter, UK**
- **magpi.cc/GvtEgf**

A monthly meeting for everyone interested in all things computers, microcontrollers, robotics, and making.

06. Preston Raspberry Jam

- Saturday 6 April
- **Fulwood Library, Preston, UK**
- **magpi.cc/gbJPhd**

A free digital making event for all ages and skill levels. Beginners welcome.

07. Cornwall Tech Jam

- Saturday 13 April
- **Cornwall College, Redruth, UK**
- **cornwalltechjam.uk**

For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming.

08. Egham Raspberry Jam

- Sunday 28 April
- **Gartner UK HQ, Staines-upon-Thames, UK**
- **magpi.cc/KZaDWu**

An opportunity to share what you have built, as well as learn from others.

FULL CALENDAR

Get a full list of upcoming events for April and beyond here:

rpf.io/jam

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area?
Want to start one?
Email Ben Nuttall about it:
jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?

Raspberry Jam advice:

Catering

"We're not allowed food and drink inside our venue. The event starts at 1pm, so we advise people to eat before they arrive, and we point out some local cafés."

Andrew Oakley – Cotswold Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **magpi.cc/2q9DHfQ**

Your Letters

Reprints

I have noticed that some of your back issues are sold out and have been for some time. I was wondering if you have any plans to print any more (I refer specifically to issue 71). Unfortunately I missed a copy as it was published between my subscriptions whilst I was on holiday and unaware my subscription had lapsed.

Paul via email

It is ultra-rare that any magazine gets reprinted – we did a reprint of issue 40 due to its popularity, but that was a very special case. You can get a free PDF of any back issue on our website and print it for your own personal use. Every now and then we do find some old print stock, so give it a periodic check to see if it's shown up. Or check out eBay in case anyone has KonMari'd their collection.

FREE RASPBERRY PI WITH THIS ISSUE
The MagPi
The official Raspberry Pi magazine
Issue 40 Christmas 2015
raspberrypi.org/magpi
#PIZERO
Raspberry Pi Zero is a real $5 computer. Learn to code while you play Minecraft (or do just about anything else you can dream up)
RASPBERRY PI ZERO
FREE WITH ISSUE #40 The MagPi
PLUS: ASTRO PI IS ENROUTE TO THE INTERNATIONAL SPACE STATION

▲ The reprint of issue 40 was done under exceptional circumstances

Making Month

What is this Month of Making all about? I keep seeing it over Twitter. Is it anything to do with Makers Hour or Makevember?

Robert via Twitter

If you managed to miss last issue, we decided to launch a little event for the Raspberry Pi and wider maker community called #MonthOfMaking. Just a call to post what you're working on, with images, to Twitter! You can see some of the fun stuff people have made in This Month in Raspberry Pi on page 86, and we'll hopefully have a lot more to show off next month in the magazine! Of course, you can also make stuff in April or any other month of the year and post it to Twitter. However, we wanted everyone to get involved this month!

▲ Make whatever you want in March!

Contact us!

- Twitter @TheMagPi
- Facebook magpi.cc/facebook
- Email magpi@raspberrypi.org
- Online raspberrypi.org/forums

Coral availability

I contacted Mouser (mouser.com) about the Coral as it seems like France is one of the only countries where the latest Coral TPU from Google is not available for purchase. I had my hopes high with the latest *The MagPi*, issue #79, and was wondering why I couldn't buy it!

Pierre-yves via Twitter

It looks like there's a restriction in place currently for France, according to Mouser, but they seem confident this will change and be available for purchase there. Otherwise, you may have to find a friend in a country that can buy it to send one on to you. Or take a quick trip on the Eurostar.

Dash button

It looks like the Amazon Dash button has been discontinued, which is a big shame! I know a few tutorials you've done incorporate it, so I was wondering if you knew of any alternatives to the Dash button for this kind of… easily hackable, quick button-press device?

Dil via the forum

To give a little insight into what it's like to write a print magazine: in the previous issue, we had recommended the Dash button for use in #MonthOfMaking. The day after the issue hit stores, the discontinuation was announced. Which is a touch frustrating! Anyway, as for replacements, some folks have already been making clones of the Dash button with ESP8266 IoT boards, and we'd hazard a guess that they'd generally be easier to use, even if they require a little more construction time. You can read up on these kind of clones here: **magpi.cc/igbPGq**.

▼ Goodbye, Dash buttons. We never really used you for your intended purpose anyway

The MagPi ESSENTIALS
LEARN | CODE | MAKE

OUT NOW IN PRINT
ONLY £3.99
from
store.rpipress.cc

The MagPi ESSENTIALS From the makers of the official Raspberry Pi magazine

GET THEM DIGITALLY:

Available on the App Store

GET IT ON Google Play

HackSpace

TECHNOLOGY IN YOUR HANDS

THE NEW MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE #17

OUT NOW

hsmag.cc

Available on the App Store

GET IT ON Google Play

WIN One of Five Keybow mechanical keyboards for your Pi

We reviewed the Keybow in issue 79 (**magpi.cc/79**) and thought it was pretty cool, especially in the way it looks.

"Keybow is a Raspberry Pi-powered mini mechanical keyboard, with twelve illuminated hot-swap keys, clear keycaps, and awesome customisable layouts and macros."

We have five Keybows to give away.

In association with PIMORONI

Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/MytktH

Terms & Conditions

Competition opens on **27 March 2019** and closes on **25 April 2019**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

NEW

Wireframe

Join us as we lift the lid on video games

Visit **wfmag.cc** to learn more

THE MAGPI #81
ON SALE 25 APR

Plus!

Hack party poppers
Automate your garden
Using GPIO Zero 1.5
Learn artificial intelligence
The 10 best laptop projects

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.org

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.org

Features Editor
Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors
Phil King and Nicola King

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Harriet Knight

Illustrator
Sam Alder

CONTRIBUTORS

Mike Cook, David Crookes, PJ Evans, Dave Honess, Nicola King, KG Orphanides, Dan Powell, Mark Vanstone

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org
+44 (0)7904 766523

Director of Communications
Liz Upton

CEO
Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Raspberry Pi Press
Mann Enterprises, Unit E,
Brocks Business Centre,
Haverhill, CB9 8QP

To subscribe
magpi.cc/subscribe

To get help:
rpipresshelp@raspberrypi.org

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.

Sound advice

How one audio artist ended up supporting Code Clubs. By **Dan Powell**

When I was twelve, my parents bought me a ZX81 for Christmas. Those of you of a similar age may remember the restrictions of coding with 1kB of RAM, although mostly I just copied lines of BASIC from the pages of Sinclair User, especially once I got my 16kB RAM pack. Friends and I would occasionally try to rewrite some of the code, with varying degrees of success, and I once tried (unsuccessfully) to learn Forth on the Acorn Electron I had upgraded to. Once I had my Computer Science O-level, my computing education stopped.

> I have an amazing set of colleagues who are able to help

I ended up in the arts, but my basic understanding of coding meant I saw a computer as a creative tool, and that gave me the confidence to use computers in my work as a sound artist. One of the challenges for me as an artist was finding a piece of software which was flexible enough for me to improvise with in a live situation with other musicians.

After trying Fruityloops, Ableton Live, and AudioMulch, I was talking to a friend in the early 2000s, who suggested I try Pure Data (Pd) – **magpi.cc/CDEFbX** – a visual programming language for creating audio and multimedia. I immediately felt slightly intimidated by the phrase 'programming language', I wondered if I had the skills to code my own music? But my basic understanding of coding gave me the confidence to dive in, and pretty soon I was writing my own instruments, effects, and building some basic interfaces with the support of the Pd community.

Code Club community

As well as being a sound artist, I work as a Programme Manager for the Raspberry Pi Foundation on Code Club. In my role supporting the Code Club regional team, my work includes building and supporting the amazing community of Code Club volunteers and the people who host the clubs. During my time at the RPF, I've met many Code Club volunteers and members of the Raspberry Pi community who, like me, don't have a lot of programming experience but learnt just enough for it to be useful in whatever they are doing in their daily lives. They now share that with young people across the world at Code Clubs, CoderDojos, and Raspberry Jams.

One of the great things for me about working at the RPF is that I have an amazing set of colleagues who are able to help with whatever music project I'm working on. Ben Nuttall, one of the Community Managers in Youth Partnerships, was very helpful when I was trying to write some Python to pass control signals from a rotary potentiometer to Pure Data, and I got stuck.

The RPF supports this by running a regular Maker Day for staff. Once a month we get together to code, build, and make. Teams across the various offices, and remote workers, take part either in person or via a Hangout. Sometimes I try out a project that the content team are working on and would like some feedback for. Lately I've been trying to learn how to design PCBs, and doing that at Maker Day means I have been able to get advice from people with far more experience.

My journey in digital making started 40 years ago. If my parents hadn't bought me that ZX81, and if I hadn't had the support of family, friends, colleagues, and the maker community, I wouldn't be writing this. So, thank you to everyone who's helped me along the way.

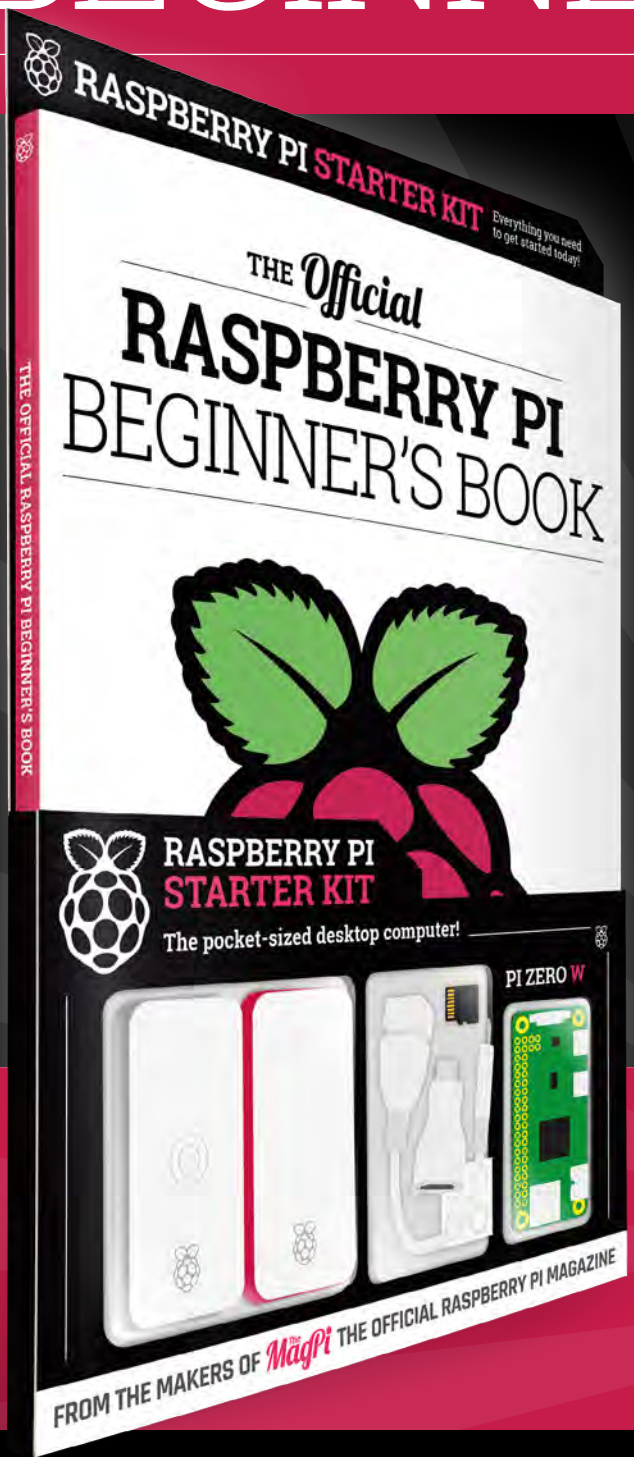
AUTHOR

Dan Powell

Programme Manager, Code Club. Brighton-based sound artist, allotment holder, and dad.

@danielpowell
soundcloud.com/dan-powell

THE *Official* RASPBERRY PI BEGINNER'S BOOK


RASPBERRY PI STARTER KIT
Everything you need to get started today!
THE Official RASPBERRY PI BEGINNER'S BOOK
THE OFFICIAL RASPBERRY PI BEGINNER'S BOOK
RASPBERRY PI STARTER KIT
The pocket-sized desktop computer!
PI ZERO W
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

LEARN COMPUTING THE EASY WAY!

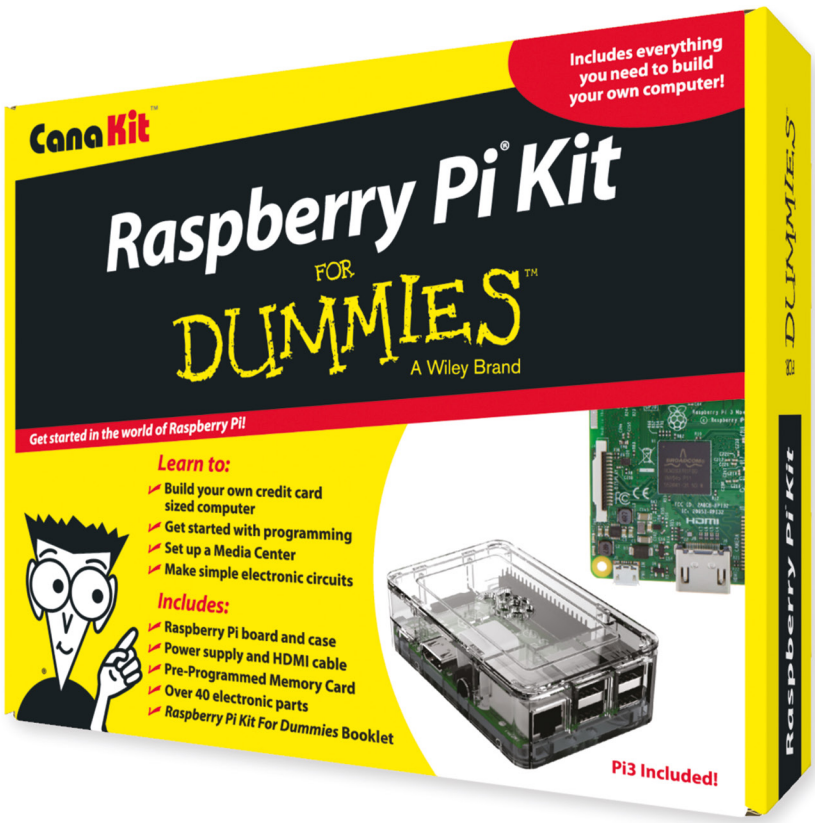
Includes

- Pi Zero W computer
- Official case with three covers
- USB and HDMI adapters
- 8GB microSD card
- 116-page beginner's book

Available now

Buy online: magpi.cc/store

CanaKit™


CanaKit
Includes everything you need to build your own computer!
Raspberry Pi® Kit
FOR DUMMIES™
A Wiley Brand
Get started in the world of Raspberry Pi!
Learn to:
Build your own credit card sized computer
Get started with programming
Set up a Media Center
Make simple electronic circuits
Includes:
Raspberry Pi board and case
Power supply and HDMI cable
Pre-Programmed Memory Card
Over 40 electronic parts
Raspberry Pi Kit For Dummies Booklet
Pi3 Included!
Raspberry Pi Kit
FOR DUMMIES

FOR DUMMIES® A Wiley Brand

Kit Includes:

- *Raspberry Pi For Dummies* Booklet
- Raspberry Pi 3 Board
- Memory Card
- Plastic Case
- 2.5A Power Supply
- HDMI Cable
- Resistors
- LEDs
- Push Button Switches
- Prototyping Breadboard
- Jumper Wires
- Heat Sinks

Available for worldwide shipping at:

WWW.CANAKIT.COM

Available in Europe through RS Components

$89.99 US DOLLARS

£69.99 EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation. For Dummies and the Dummies Man logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Used under license. RS logo is a registered trademark of RS Components Ltd. CanaKit is a registered trademark of Cana Kit Corporation.

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO